

Oscar Nordman

# Linja-auton polttoaineenkulutuksenmittaus- järjestelmän suunnittelu ja toteutus

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Auto- ja kuljetustekniikka

Insinöörityö

28.11.2014

Tekijä(t) Otsikko  Sivumäärä Aika	Oscar Nordman Linja-auton polttoaineenkulutuksenmittausjärjestelmän suunnittelu ja toteutus  41 sivua + 4 liitettä 28.11.2014
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Auto- ja kuljetustekniikka
Suuntautumisvaihtoehto	Autosähkötekniikka
Ohjaaja(t)	Korjaamopäällikkö Esa Syrjänen Autoelektroniikan lehtori Vesa Linja-aho
<p>Opinnäytetyössä suunniteltiin ja rakennettiin Veolia Transport Finland Oy:lle yhteen yrityksen linja-autoista tiedonkeruujärjestelmä, joka kerää kattavasti dataa koeajettaessa yhtiölle uusia vieraita linjoja linjakilpailutusta varten. Työn tärkein tavoite oli saada tietoon mahdollisimman tarkka polttoaineenkulutus tietyllä ajettavalla linjalla. Tämän vuoksi oli kannattavinta integroida tiedonkeruujärjestelmä yhteen linja-autoista ja simuloida sillä oikeata ajotilannetta (pysäkeille pysähtymiset, matkustajien massa, ajotavan vaikutus jne.) mahdollisimman tarkkaan.</p> <p>Työ alkoi suunnittelulla ja hankkimalla osia, joista suurin osa löytyi Teknologia 13 -messuilla, Messukeskuksesta. Polttoaineen kulutuksen mittaus päätettiin toteuttaa virtausanturilla, jonka data kerättäisiin dataloggerin avulla suoraan tietokoneelle. Ratkaisu oli edullinen ja suhteellisen helppo toteuttaa. Lisäksi virtausantureita ja dataloggereita on markkinoilla useita kymmeniä, joten omaan käyttötarkoitukseen on helppo löytää sopivat laitteistot. Itse polttoaineenkulutuksenmittausohjelma suunniteltiin ja toteutettiin itse käyttäen hyväksi LabView 2013 -ohjelmistoa. Järjestelmä asennettiin kohteeksi valikoituun linja-autoon.</p>	
Avainsanat	Kulutuksenmittaus, Labview, virtausanturi, dataloggeri, linja-auto

Author(s) Title Number of Pages Date	Oscar Nordman Design and Implementation of Fuel Consumption Measurement System for Bus 41 pages + 4 appendices 28 November 2014
Degree	Bachelor of Engineering
Degree Programme	Automotive and Transport Engineering
Specialisation option	Automotive Electronics Engineering
Instructor(s)	Chief Workshop Manager, Esa Syrjanen Automotive Electronics Lecturer, Vesa Linja-aho
<p>The objective of this thesis was to design and build a data acquisition structure for Veolia Transport Finland Oy, which would gather information from different variables when test driving not already familiar bus routes for the firm. The first priority of the data acquisition structure was to measure fuel economy as precisely as possible. This meant that the system should be integrated to the bus itself, so that the circumstances would be quite close to the real situation.</p> <p>The project started by planning the whole system and by obtaining all the parts needed for the data acquisition system. Almost all the parts needed were found at the "Teknologia 13" exhibition. Measurement of the fuel economy was determined to be carried out with a fuel flow meter from which output signals would be gathered with a data logger straight to a hard drive of a PC. There are several data loggers and flow meters on the market, so finding the most suitable ones for this project was quite an easy task. The data acquisition program itself was decided to be engineered and actualized using the National Instruments LabView 2013 software. The complete logging system was installed in a bus chosen for this project.</p>	
Keywords	Fuel economy, Labview, flow meter, datalogger, bus

## Contents

1	Johdanto	1
2	Veolia Transport Finland Oy	2
3	DH10A-moottori	2
4	EDC	4
4.1	Yleistä	4
4.2	EDC:n tunnistimet	5
4.2.1	Kaasupolkimen asentoanturi	6
4.2.2	Moottorin nopeusanturi	6
4.2.3	Vakionopeudensäädin ja jarrupoljin katkaisin	7
4.2.4	Jäähdytysnesteen lämpötilatunnistin	8
4.2.5	Ahtoilman lämpötunnistin ja ahtopaineanturi	8
5	Polttoainejärjestelmä	9
5.1.1	Ruiskutuspumppu	9
5.1.2	Ruiskutuspumpunsäädin	10
5.1.3	Siirtopumppu	12
5.1.4	Suuttimet	13
5.1.5	Polttoainesuodatin	14
6	Projektin komponentit	14
6.1	DFM 8D virtausanturi	14
6.2	USB-6008 dataloggeri	17
6.3	NI LabView 2013	19
7	Kulutusmittauksen simulointi	20
8	Järjestelmän asennus	29
8.1	Virtausanturin asennus	29
8.2	Virtausanturin johdotukset	31
8.3	Muiden signaalien johdotukset	33
8.3.1	Matkatieto	34
8.3.2	Ulkolämpötila ja kaasupolkimen asento	34
8.4	Kulutusmittausohjelman teko ja testaus	34

9	Mittaukset ja loppupäätelmät	36
9.1	Ensimmäinen testimittaus	36
9.2	Toinen testimittaus	37
9.3	Kolmas testimittaus	37
9.4	Neljäs testimittaus	38
9.5	Loppupäätelmät	38
	Lähteet	41

#### Liitteet

Liite1. Polttoaineenkulutusmittausohjelma

Liite2. Kulutusmittausohjelman rakenne

Liite3. Kulutusmittausohjelman UI

Liite4. Kulutusmittausohjelman ensimmäinen versio

## Lyhenteet

EDC	<b>Electronic Diesel Control.</b> Sähköisesti ohjattu polttoaineen ruiskutus dieselissä.
R6	6 sylinterinen mäntämoottori, jossa sylinterit ovat vierekkäin rivissä. Käytetään myös nimitystä ”suora moottori”.
TPS	<b>Throttle Position Sensor.</b> Anturi joka tunnistaa kaasupolkimen asennon.
ESP	<b>Electronic Stability Program.</b> Ajonvakautusjärjestelmä, joka korjaa kuljettajan tekemiä ajovirheitä käyttäen hyväkseen ajoneuvon eri osalueita, kuten jarruja.
ETC	<b>Electronic Transmission Control.</b> Automaattivaihteiston ohjausjärjestelmä.
TCS	<b>Traction Control System.</b> Luistonestojärjestelmä, joka pyrkii minimoimaan tien ja renkaan välisen luiston vähentämällä moottorilta tulevaa vääntömomenttia tai jarruttamalla pyöriä.
UI	<b>User Interface.</b> On se osa ohjelmistosta tai laitteesta, jonka kautta tuotteen käyttäjä ohjelmistoa tai laitetta käyttää.

## 1 Johdanto

Opinnäytetyössä suunniteltiin ja rakennettiin Veolia Transport Finland Oy:lle yhteen yrityksen linja-autoista tiedonkeruujärjestelmä, joka kerää kattavasti dataa koeajettaessa yhtiölle uusia vieraita linjoja linjakilpailutusta varten. Työn tärkein tavoite oli saada tietoon mahdollisimman tarkka polttoaineenkulutus tietyllä ajettavalla linjalla. Tämän vuoksi oli kannattavinta integroida tiedonkeruujärjestelmä yhteen linja-autoista ja simuloida sillä oikeata ajotilannetta (pysäkeille pysähtymiset, matkustajien massa, ajotavan vaikutus jne.) mahdollisimman tarkkaan.

Linja-autoksi valikoitui vuosimallia 1998 oleva Volvo Carrus B10. Kyseessä on nk. matalattialinja-auto noin 10-litraisella R6-dieselmoottorilla. Auto on suunniteltu nimenomaan kaupunkiliikennettä varten. Kyseessä on sen verran vanha linja-auto, ettei siinä ole vielä minkäänlaista väylätekniikkaa vaan alustan ja korin välinen tiedonsiirto on täysin analogista. Moottorin ruiskutusjärjestelmä on kuitenkin jo EDC-ohjattu. Projektin autoksi valittiin vanhempi auto, koska tarkkaa tiedonkeruuta varten auton alkuperäistä rakennetta oli muutettava sen verran, ettei sillä olisi enää ollut kannattavaa ajaa normaalia linja-ajoa. Lisäksi auton käyttöikä linjaliikenteessä alkoi olemaan täynnä (16 vuotta), joten nk. tiedonkeruubussista haluttiin tehdä pysyvä ratkaisu.

Polttoainemittauksen tapaa ja laitteistoa mietittiin pitkään, sillä lähestymistapoja asiaan oli monia. Loppujen lopuksi päädyttiin käyttämään virtausanturia, joka laskisi rivipumpulle tulevan ja lähtevän (ylimääräisen, takaisin polttoainetankkiin palaavan) polttoaineen erotuksen. Itse tiedonkeruujärjestelmä päätettiin rakentaa PC:n ympärille. Virtausanturin sekä muu haluttu data kerättiin talteen dataloggerilla, joka siirrettiin USB-väylän avulla kannettavan tietokoneen kovalevyille. Raakadatan hallinnoimista varten loin National Instrumentsin Labview 2013 -ohjelmalla käyttöliittymän, joka lukisi ja muuntaisi raakadatan suoraan luettavaan muotoon. Esimerkiksi virtausanturin lähettämän digitaalisen signaalin suoraan polttoainekulutukseksi litraa / 100 kilometriä.

## 2 Veolia Transport Finland Oy

Työn tilaajana oli Veolia Transport Finland Oy. Se on osa monikansallista liikenneyhtiötä Veolia Transdeviä. Transdev -konserni on alun perin ranskalainen, ja se tuottaa moderneja maa-, raide- ja vesiliikenteen matkustuspalveluja kaikkiaan 21 eri maassa. [1.]

Suomessa vuonna 1994 Axel Johnson -konsernin omistama Linjebuss-yhtiö osti sekä Vantaan Liikenne Oy:n että Westendin Linja Oy:n linjaliikenteen ja kaluston. Kolme vuotta myöhemmin eli vuonna 1997 ranskalainen pörssiyhtiö CGEA osti Linjebuss -konsernin, jolla oli tuohon aikaan liiketoimintaan pohjoismaissa ja lisäksi Saksassa sekä Belgiassa. Vuonna 1999 perustettiin CGEO Transport, jonka nimi kuitenkin vaihtui jo vuoden päästä Connex Transport AB:ksi. Vuonna 2007 nimi muuttui jälleen, tällä kertaa Veolia Transport Finlandiksi. [2.]

Nykyään Veolialla on viisi eri varikkoa:

- Veolia Transport Espoo Oy, Suomenoja
- Veolia Transport Vantaa Oy, Tuupakka ja Hakunila
- Veolia Transport Helsinki Oy, Koskela ja Tattarisuo

Korjaamotoimintaa on kolmella eri varikolla Suomenojalla, Tuupakassa ja Hakunilassa.

Työntekijöitä konsernissa on yhteensä hieman alle 700 henkilöä ja linja-autoja yhteensä melkein 350 kappaletta. Liikevaihto vuonna 2009 oli n. 39 miljoonaa euroa. [1.]

## 3 DH10A-moottori

Projektiin valitun Volvo Carrus B10:nen voimanlähde on DH10A-moottori. Se on R6 eli 6-sylinterinen rivimoottori. Moottori perustuu vanhempaan THD104-moottoriin ja DH10A:ssa onkin sen tunnusomaisia piirteitä, kuten vaakatasoon kallistettu sylinteriryhmä, turboahdin sekä 9,6 litran iskutilavuus. Moottorissa on säädettävä ahtopaine ja ahtoilman vesijäähdytys, jotka parantavat sen taloudellisuutta ja antavat laaakeamman vääntökuvaajan. [3, s. 2.] Alla olevasta taulukosta 1 näkyy DH10A-moottorin suorituskäytiedot.



Taulukko 1: DH10A 245 -moottorin perustiedot [3, s. 4].

Perustiedot	
Mallimerkintä	DH10A 245
Nettoteho (kW)	180
Käyntinopeudella (rpm)	2000
Max. Momentti (Nm)	1050
Käyntinopeudella (rpm)	1050

Verrattuna THD104:een DH10A:n männänpäät on muotoiltu uudelleen, mutta ovat edelleen ”re-entry”-tyyppiset. Ruiskutusjärjestelmä on elektronisesti ohjattu (EDC). Kehitetyn ruiskutuspumpun ansiosta ruiskutuspainetta on nostettu ja uusilla suuttimilla palamista on saatu tehostettua, minkä ansiosta moottori on taloudellisempi ja saastuttaa entistä vähemmän. Taulukossa 2 näkyy DH10A:n tekniset arvot.

Taulukko 2: DH10A 245 moottorin tekniset arvot [3, s. 4] .

Tekniset arvot	
Sylinteriluku	6
Sylinterin halkaisija (mm)	120,65
Iskunpituus (mm)	140
Iskutilavuus (l)	9,6
Puristussuhde	20:1
Ruiskutusjärjestys	1-5-3-6-2-4
Hidas tyhjäkäynti (rpm)	530
Nopea tyhjäkäynti (rpm)	2200
Moottorin kokonaispaino (kg)	960

## 4 EDC

### 4.1 Yleistä

EDC eli Electronic Diesel Control on sähköinen ohjaus- ja valvontajärjestelmä polttoaineenruiskutusta varten. Järjestelmä toimii siten, että moottorissa ja ohjaamossa on joukko antureita (esim. kaasupolkimen asentoanturi, ahtopaine anturi), jotka välittävät erilaisia signaaleja elektroniseen ohjausyksikköön. Ohjausyksikkö käsittelee tulevat signaalit ja niiden perusteella ohjaa ruiskutuspumpun toimintaa eli mm. polttoaineen ruiskutusmäärää ja ruiskutusennakkoa. EDC:n avulla pystytään mukautumaan paremmin erilaisiin ajotilanteisiin, jolloin moottorin hyötysuhde ja täten myös taloudellisuus paranee. Autoissa, joissa on EDC, ei kaasupolkimen asento pelkästään määritä sylinteriin ruiskutettavaa polttoaineen määrää, vaan suihkutussmäärään vaikuttaa lukuisat muuttujat. Niitä ovat muun muassa

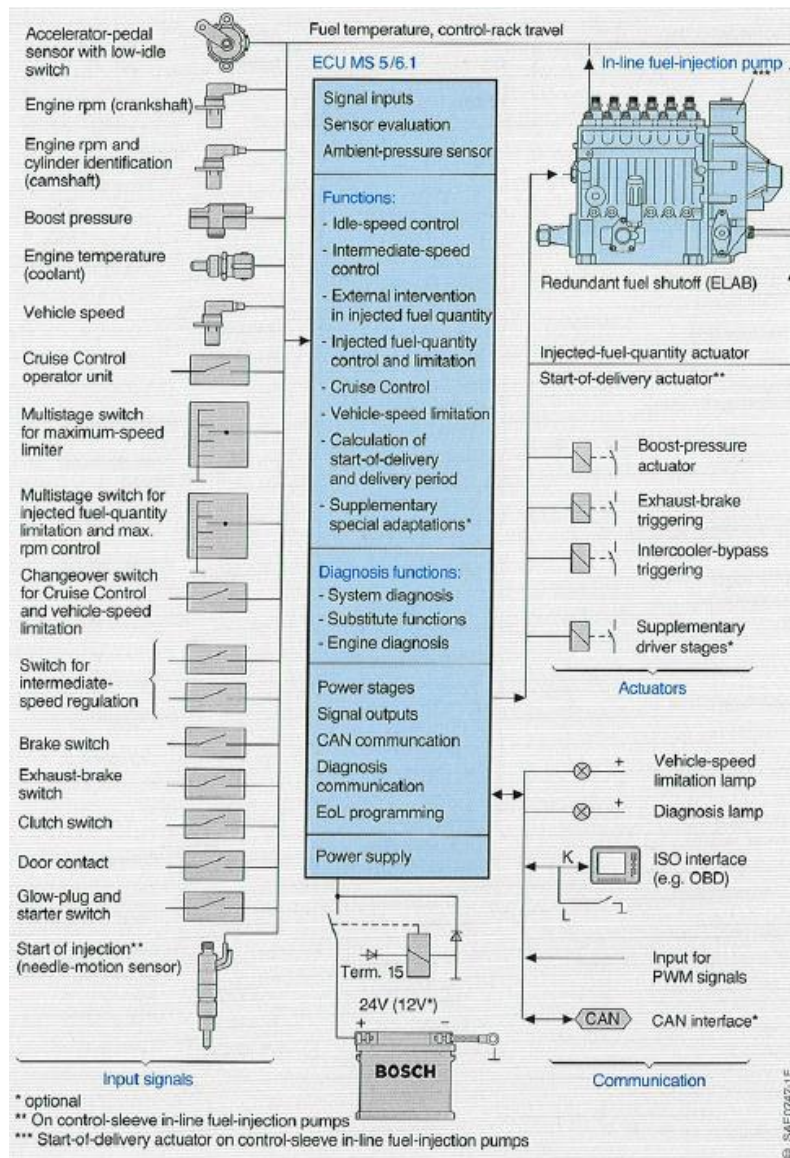
- kaasupolkimen asento
- moottorin toimintatila (tyhjäkäynti, kylmäkäynti jne.)
- moottorin lämpötila
- pakokaasupäästöihin vaikuttavat tekijät (esim. scr-regenerointi, ureajärjestelmät)
- muiden järjestelmien väliintulo (TCS, ESP jne.). [4, s. 355.]

Kaikki yllä mainitut muuttujat vaikuttavat lopulliseen sylinteriin ruiskutettavan polttoaineen määrään.

Kuten jo aikaisemmin tuli ilmi, niin EDC keskustelee myös muiden ajoneuvossa olevien elektronisten järjestelmien kuten Traction Control System (TCS), Electoctronic Transmission Control (ETC) ja Electronic Stability Program (ESP) kanssa. Tämä tarkoittaa sitä, että moottorinohjausjärjestelmä voidaan täysin integroida ajoneuvon kokonaishallintajärjestelmän kanssa. Tämä taas mahdollistaa ominaisuudet, kuten moottorin väännön alentamisen automaattivaihteiston vaihtaessa tai väännön alentamisen pyörien menettäessä pitoa.

## 4.2 EDC:n tunnistimet

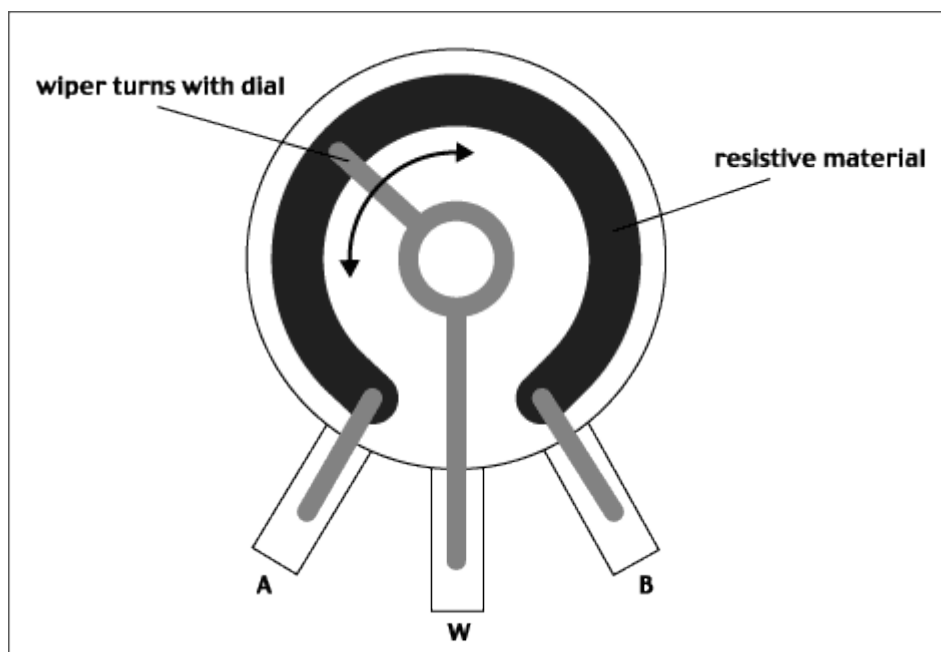
Jotta EDC pystyisi ohjaamaan moottorin toimintaa tarkoituksen mukaisesti, tarvitsee se tietoa moottorin ja muiden ajoneuvon järjestelmien tilasta sekä ulkopuolisista muuttujista. EDC on integroitu kokonaisajoneuvojärjestelmään, jolloin keskustelu muiden elektronisten järjestelmien kanssa on mahdollista. EDC:n tiedonkeruuta varten on ajoneuvoissa kymmeniä antureita, jotka mittaavat eri suureita ja muuttavat ne sähköisiksi signaaleiksi, joita EDC ymmärtää ja pystyy hallinnoimaan. Kuvassa 1 on raskaan kaluston EDC:n komponentit ruiskutuspumppuineen.



Kuva 1. Raskaan kaluston dieselmootorin EDC:n ohjaukseen vaikuttavat komponentit [4, s. 357].

#### 4.2.1 Kaasupolkimen asentoanturi

Kaasupolkimen asentoanturi (TPS – Throttle Position Sensor) on anturi, joka kertoo nimensä mukaisesti kaasupolkimen asennon. Vanhemmissa ajoneuvoissa se on toteutettu potentiometrillä ja uudemmissa käyttäen Hall-efektiä tai induktiivista sensoria. Potentiometrillä toteutetussa versiossa kaasupolkimelta lähtee suora vipu anturille, joka säättää potentiometrin sisällä olevia 2–3:a harjaa, jotka liukuvat resistiivistä nauhaa pitkin muuttaen näin potentiometrin vastusarvoa. Hall- tai induktiivisella anturilla toteutetut asentoanturit tunnistavat polkimen asennon magneettikentän muutosta hyväksi käyttäen. Tämän etu on se, että koskettavia osia ei ole ja ainakin teoriassa osan pitäisi olla pitkäikäisempi.

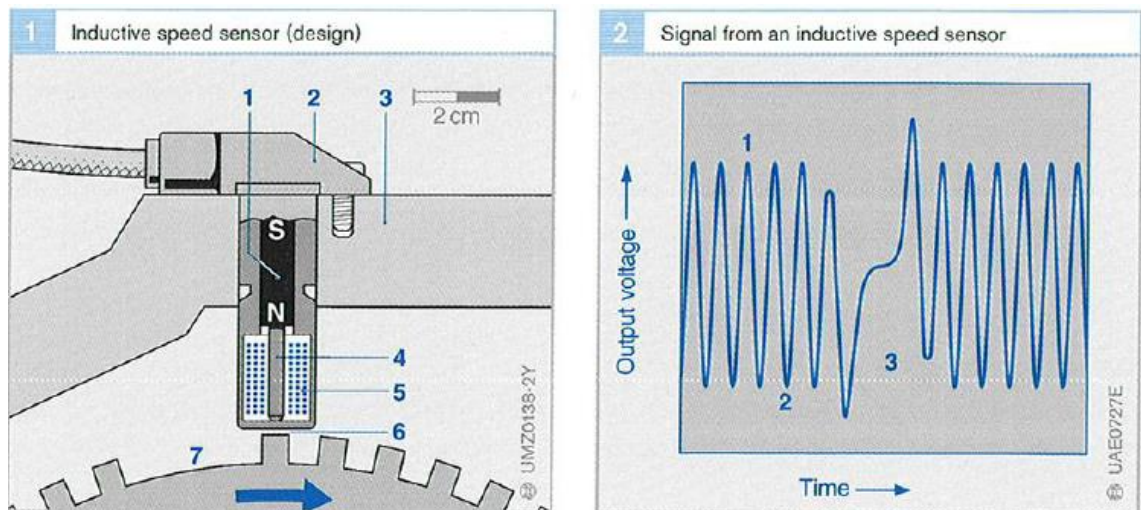


Kuva 2. Potentiometri.

#### 4.2.2 Moottorin nopeusanturi

Anturin tehtävänä on mitata moottorin nopeutta ja määrittää kampiakselin asento. Lisäksi nokka-akselilla oleva anturi mittaa myös moottorin vaihetta.

Moottorin nopeus antureitakin on monenlaisia, mutta ylivoimaisesti yleisin on induktiivisesti toteutettu malli. Anturi on sijoitettu suoraan ferromagneettisen liipaisin pyörän eli vauhtipyörän yläpuolelle jättäen niiden väliin pienen ilmaraon. Vauhtipyörässä on hampaita tasaisin välimatkoin kuten kuvasta 3 näkyy.



Kuva 3. Induktiivisen nopeusanturin rakenne ja signaali [4, s. 418].

Vauhtipyörän pyöriessä ja ilmaraon muuttuessa muuttuu myös liipaisinpyörän ja anturin välissä oleva magneettikenttä, joka saa aikaan kuvan 3 oikealla puolella olevan signaalin. Induktiivinen anturi pyrkii vastustamaan magneettikentän muutosta ja näin olleen anturiin indusoituu jännite. Jokainen piikki siis kuvastaa yhtä vauhtipyörässä olevaa hammasta. Kuvissa näkyvä 2–3 hampaan rako on kampiakselin asennon tunnistusta varten.

#### 4.2.3 Vakionopeudensäädin ja jarrupoljin katkaisin

Vakionopeudensäädintä käytetään kolmeen eri toimintoon:

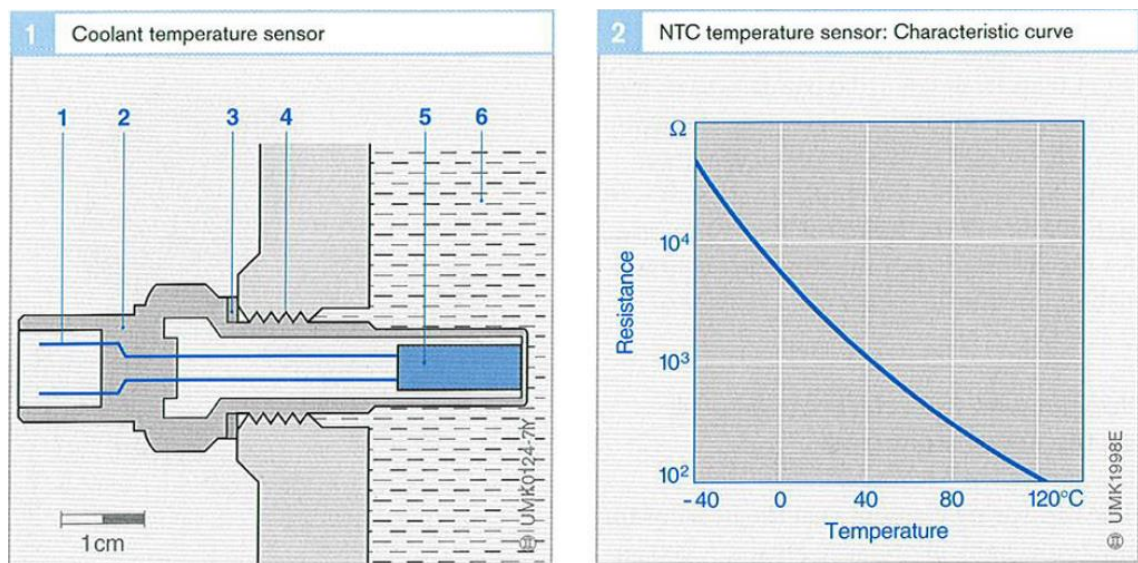
- tyhjäkäynnin säätö
- nopean tyhjäkäynnin säätö
- vakionopeuden säätö.

Vakionopeudensäätimen toiminnan katkaisee jarrupolkimen katkaisin jarruttaessa tai hidastettaessa pakokaasujarrun avulla.

#### 4.2.4 Jäähdytysnesteen lämpötilatunnistin

Lämpötilatunnistimen avulla EDC-ohjausyksikkö tunnistaa jäähdytysnesteen lämpötilan ja käyttää tietoa säätääkseen täyden kuorman ruiskutusmäärää, hidasta tyhjäkäyntiä, rajakäyntinopeutta ja käynnistettäessä rikastuksen määrää.

Jäähdytysnesteen lämpötila-anturina käytetään yleisesti NTC-termistoria eli vastusta, jonka resistanssi muuttuu lämpötilan muuttuessa. NTC:n lämpötilakerroin on negatiivinen joka tarkoittaa sitä, että anturin resistanssi pienenee lämpötilan kasvaessa, kuten kuvan 4 oikean puoleisesta kuvaajasta näkee.



Kuva 4. NTC tyyppisen lämpötila-anturin rakenne ja toimintakuvaaja [4, s. 413].

#### 4.2.5 Ahtoilman lämpötunnistin ja ahtopaineanturi

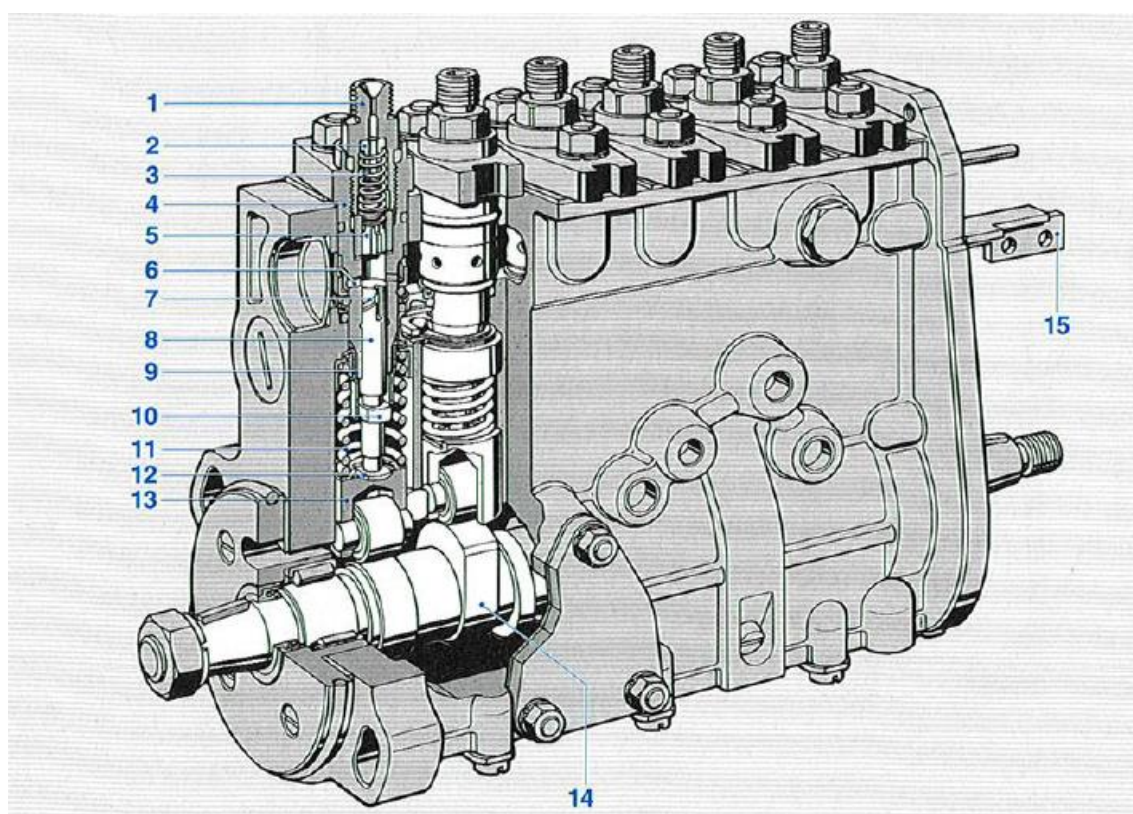
Kertoo EDC:lle tiedon ahtoilman lämpötilasta ja tarvittaessa pienentää polttonesteen ruiskutusmäärää silloin, kun ahtoilman lämpötila on tavallista korkeampi. Ahtopaineanturi kertoo EDC:lle ahtopaineen ja se vaikuttaa ruiskutettavaan polttoainemäärään. Ahtopaineanturi vertailee imusarjan painetta referenssipaineeseen, joka ei kuitenkaan ole ympäröivä ulkoilman paine.



## 5 Polttoainejärjestelmä

### 5.1.1 Ruiskutuspumppu

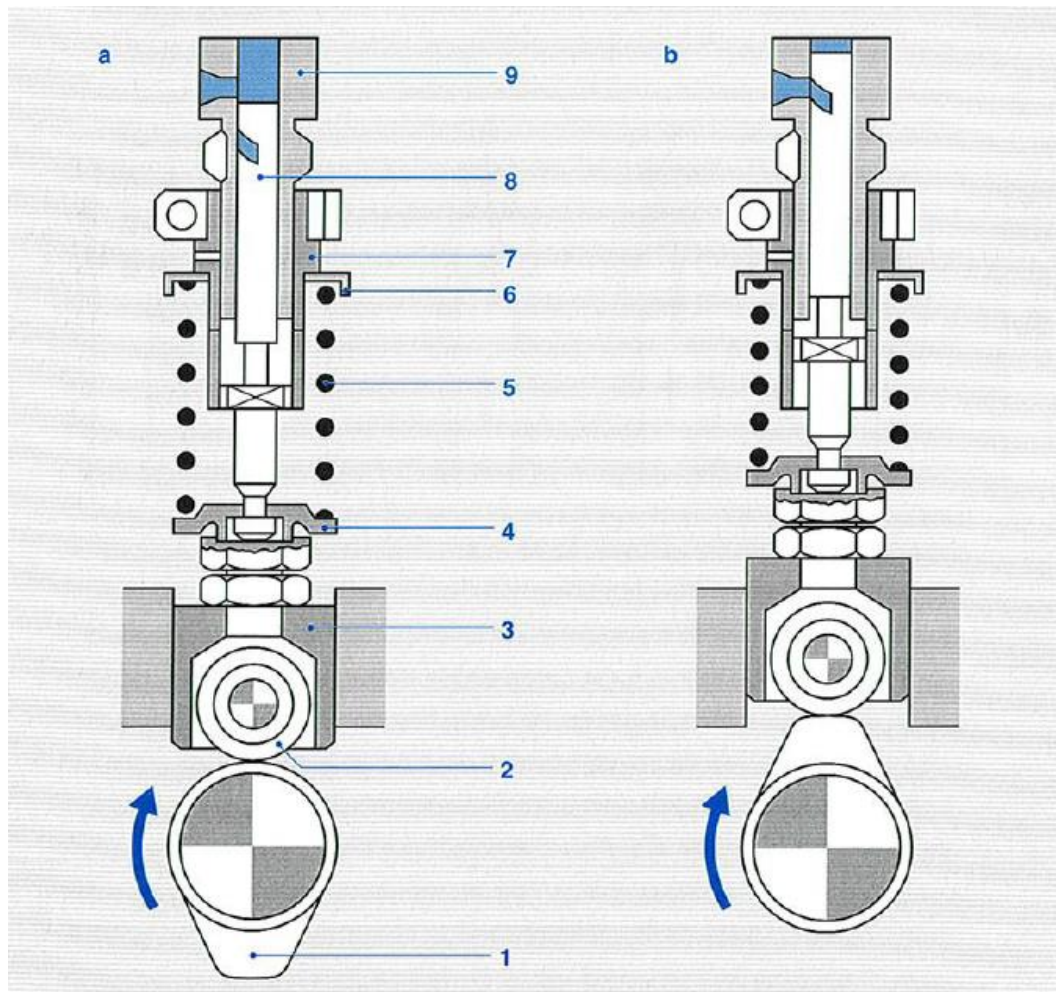
DH10A:n polttoaineen ruiskutuksesta vastaa rivipumppu tyyppinen ruiskutuspumppu (kuva 5). Pumpun rakenteelle ominaista on oma nokka-akseli (14), joka saa voimansa yleensä suoraan moottorin jakopäästä rattaiden välityksellä. Pumpun perusrakenteeseen kuuluu mäntä (8), pumpun sylinteri (4), säätötanko (9), helix (7), männän jousi (11), jousilautanen (12), rullapyörä (13) ja nokka-akseli.



Kuva 5. Rivityypinen ruiskutuspumppu [4, s. 103].

Seuraavalla sivulla olevassa kuvassa 6 näkyy pumpun toimintamalli. A-kohdassa pumppu on BDC-asemassa eli mäntä on alakuolokohdassa. Nokka-akseli ei pyri nostamaan mäntää ylöspäin ja männän jousi painaa mäntää jousivoimallaan alaspäin. BDC:ssä ollessaan ulkoinen polttoaineen siirtopumppu täyttää pumpun sylinteriä eikä suuttimille tule painetta eli ruiskutusta ei tapahdu. Kohdassa B nokka-akseli on pyörähtänyt 180 astetta ja nokka-akselin nokka on nostanut männän yläkuolokohtaansa kumoten männänjousen voiman. Tätä kutsutaan TDC-asemaksi.

Männän noustessa ylöspäin se on sulkenut sylinteritäytön linjan, paineistanut polttoaineen ja lopuksi yläkuolokohdassa vapauttanut paineen paluulinjaa pitkin. Polttoaineen paineistuessa tarpeeksi on rivipumpun yläpäässä olevat paineventtiilit avautuneet ja päästäneet polttoaineen paineella suuttimille. Paluulinjan auetessa ja paineen laskiessa paineventtiilit sulkeutuvat jälleen ja polttoaineen suihkutussuuttimissa lakkaa.



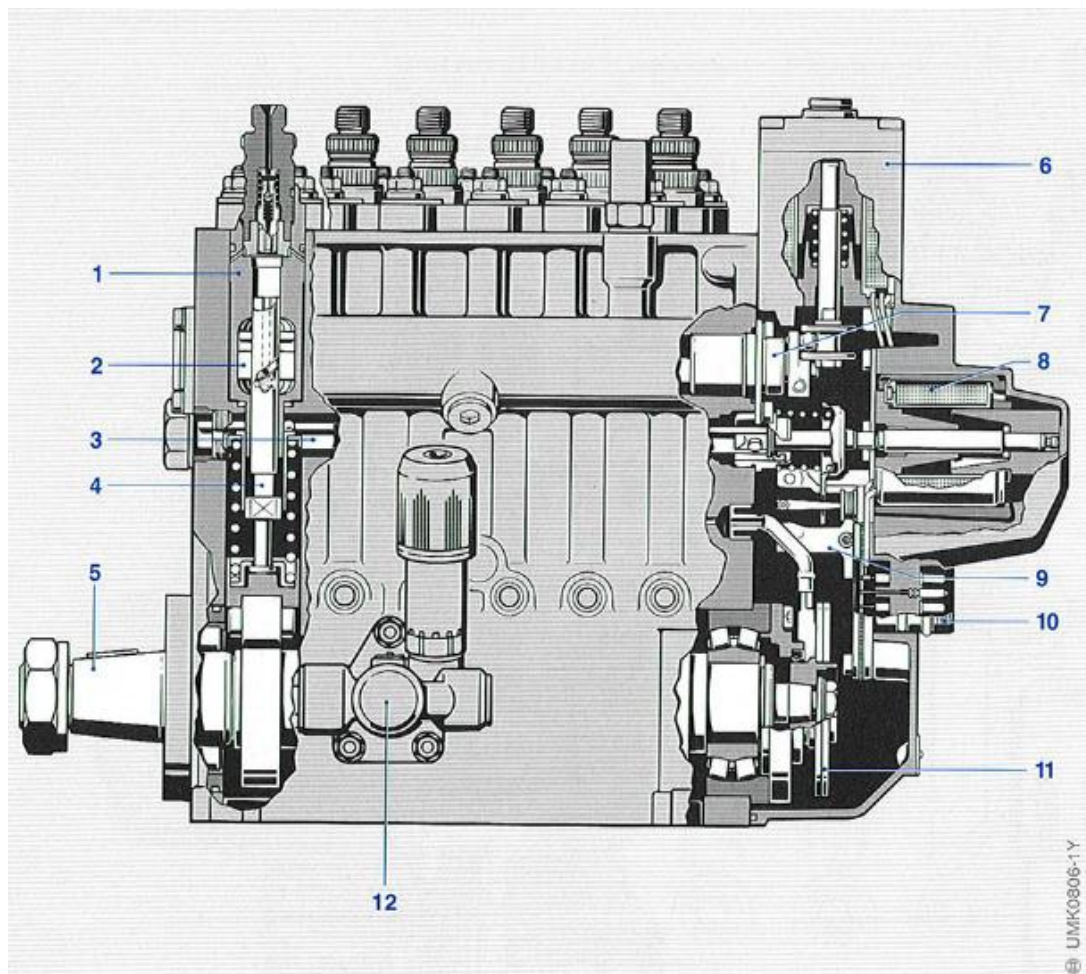
Kuva 6. Ruiskutuspumpun toimintakuva [4, s. 105].

### 5.1.2 Ruiskutuspumunsäädin

Säädinyksikkö on asennettu rivipumpun (kuva 7) toiseen pätyyn säädinkoteloon. Säätimessä on voimakas sähkömagneetti (8), joka vaikuttaa ruiskutuspumpun säätötangon (3) asentoon. Säätötanko pyörittää ruiskutus-pumpussa olevia mäntiä (4), joissa on helix poraus ja joka vaikuttaa polttoaineen syötön määrään. Säätötankoa



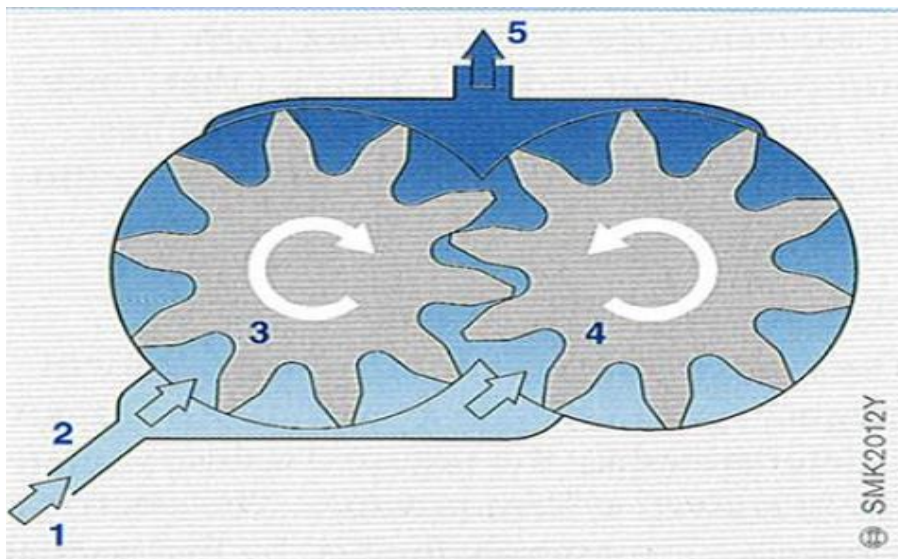
liikuttavaa sähkömagneettia ohjaa EDC siihen tulevien signaalien perusteella, joita ovat mm. kaasupolkimen asento, ahtopaine- ja lämpötilatieto ja moottorin pyörintänopeus. DH10A-moottorin pääkäyntinopeusanturi on sijoitettu ruiskutuspumpun säätimeen ja se tunnistaa käyntinopeuden ruiskutuspumpun nokka-akselilla olevasta hammaspyörästä. Ruiskutusennakkoa säädetään säätämällä säätöholkin (2) asentoa. Holkin asento määrää, missä männän asennossa polttoaineen paineistaminen alkaa (eli ruiskutus) ja milloin mäntä saavuttaa spill-portin eli milloin ruiskutus lakkaa.



Kuva 7. Rivipumpun läpileikkauskuva [4, s. 146].

### 5.1.3 Siirtopumppu

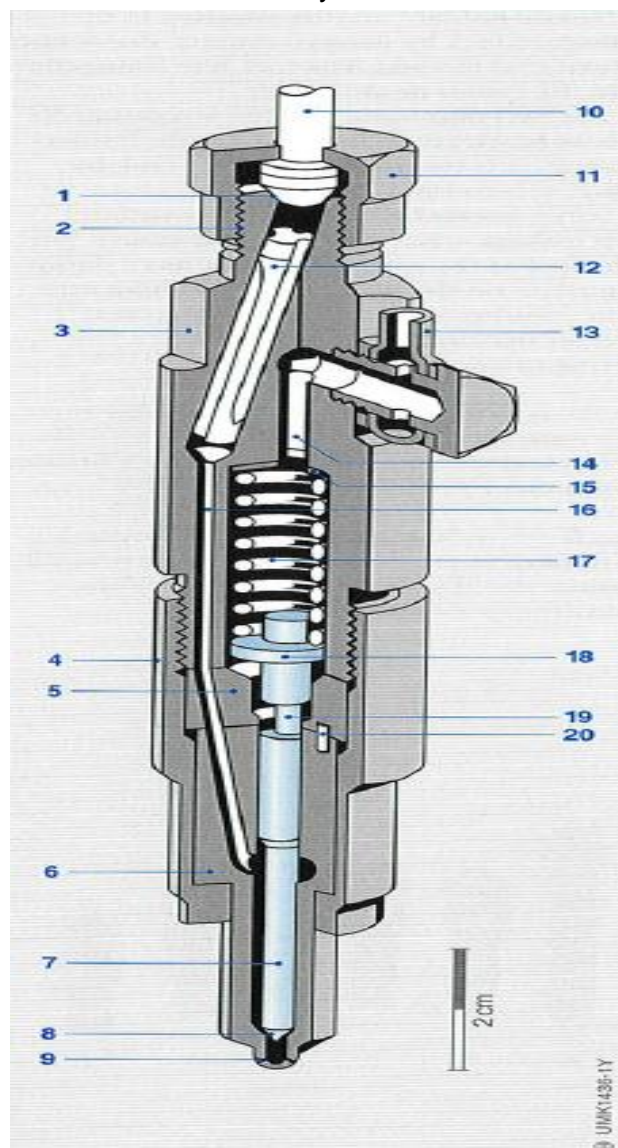
Siirtopumpun tehtävänä on imeä polttoainetta tankista ja kuljettaa sitä jatkuvasti korkeapainepumpulle. Aksiaalisissa ja radiaalisissa ruiskutuspumpuissa siirtopumppu on integroitu ruiskutuspumppun yhteyteen, mutta muissa järjestelmissä se on ruiskutuspumppusta erillinen. Normaalisti henkilöautoissa siirtopumppu on sijoitettu suoraan tankkiin ja se toimii sähköisesti. Raskaalla puolella on yleistä, että siirtopumppu on ruiskutuspumppusta tai tankista erillinen elementti. Yleisesti ottaen ne ovat mekaanisella yhteydellä toimivia pumppuja. Siirtopumpun voima otetaan rattaiden välityksellä joko korkeapainepumpulta tai suoraan jakopäästä. Kuvassa 8 näkyy mekaanisesti toimiva rataspumppu. Polttoaine tulee imuputkea (1) pitkin ja rattaat (3) ja (4) paineistavat polttoaineen eristään imu- ja painepuolen toisistaan ja samalla estäen, ettei polttoaine pääse virtaamaan takaisin imuputkea pitkin. Pumpun syöttö riippuu moottorin nopeudesta. Tämän vuoksi syötön ohjaus tapahtuu joko hallitsemalla imupuolen polttoaineen virtausta tai pumpun jälkeisellä ylivuotoventtiilillä.



Kuva 8. Ratastyypinen siirtopumppu [4, s. 88].

#### 5.1.4 Suuttimet

Suuttimien tehtävänä on suihkuttaa ruiskutuspumpulta tuleva polttoaine moottorin sylinteriin mahdollisimman hienojakoisena sumuna. Tämän vuoksi ruiskutuspaineet ovat hyvin korkeat, välillä 1600—2200 bar. Kuvassa 9 näkyy suuttimen rakenne. Ruiskutuspumpulta paineella tuleva polttoaine tulee suuttimeen putkea (10) pitkin ja jatkaa matkaansa suuttimen alaosaan. Painejousi (17) ja painetanko (18) painavat suuttimen neulaa (7) suutinta (9) vasten. Kun polttoneste tulee ruiskutuspumpusta polttoainekanavaa (12) pitkin suuttimen alaosaan ja lopulta voittaa neulaa alaspäin työntävän painejousen voiman, nousee neula istukastaan ja suutin avautuu. Tällöin polttoaineen suihkutusta alkaa. Kun paine polttoaine-suuttimessa jälleen laskee, painaa painejousi neulan jälleen istukkaansa, joka sulkee suuttimen ja lopettaa sumuttamisen välittömästi. Suuttimen neulaan kohdistuvan voiman määrää pelkästään painejousi. Painejousen jännitystä ja täten suuttimen avautumispainetta voidaan säätää. Kyseisessä mallissa se onnistuu säätölevyllä.



Kuva 9. Polttoainesuutin [4, s. 310].

### 5.1.5 Polttoainesuodatin

Ajoneuvossa on 2 polttoainesuodatinta ja ne ovat sijoitettu käynnistysmoottorin alapuolelle. Suodattimet ovat ”spin-on” tyyppiset ja rinnan kytketty. Suodatinpanoksissa on spiraalille kiedottua paperia, jolla on saatu aikaan erittäin suuri suodatinpinta-ala. Paperikerrokset on taitettu siten, että niistä muodostuu ylöspäin aukeavia taskuja. Polttoneste virtaa pituussuunnassa näiden läpi, jolloin suodatinpaperi erottaa polttonesteestä likahiukkaset ja veden yms.

## 6 Projektin komponentit

Projektin toteutustapaa mietittiin hyvin pitkään. Esimerkiksi polttoaineen kulutuksen mittauksen olisi voinut toteuttaa monella eri tapaa. Tarkin tapa olisi ollut mitata polttoaineen massaa, mutta käytännön toteutus olisi ollut hieman hankala. Hyvä kompromissi tarkkuuden ja käytännöllisyyden välillä oli mitata polttoaineen virtausta virtausanturilla. Kun mittauksella oli selvillä, piti vielä keksiä, kuinka kerätä kaikki haluttu data (kulutus, kaasupolkimen asento, ajonopeus jne.) talteen.

1.10.–3.10.2013 järjestettiin Teknologia 13 messut, johon osallistuin. Tuolloin keskustellessani useiden eri yhtiöiden edustajien kanssa tuli selväksi, että projektin järjestelmä olisi kannattavinta rakentaa PC:n ympärille. Päädyimme hankkimaan kannettavan tietokoneen, joka tallentaisi kaiken tarvittavan datan dataloggerin avulla. Dataloggeriksi valikoitui NI:n USB-6008-loggeri, jonka ominaisuudet ja hinta miellyttivät. Dataa käsitteleviä valmiita ohjelmia olisi myös löytynyt, mutta päätin tehdä oman käyttöliittymän käyttämällä NI:n LabView 2013 ohjelmaa. Yksi selkeä etu ratkaisussa oli se, että datankeruu ohjelma oli tällöin täysin muokattavissa omiin tarpeisiin ilman muiden valmistajien rajoitteita.

### 6.1 DFM 8D virtausanturi

Virtausanturi on sveitsiläisen yrityksen Aquametrin valmistama ja Suomeen sen toimitti Sarlin Oy. Anturin perusperiaate on mitata tilavuudenmuutostansa siten, että aina kun anturin sisäinen tilavuus eli 12.44ml on syrjäytetty, lähettää se maadoitus pulssin,

kuten kuvan 11 oikean puoleiselta kuvaajalta voidaan havaita. Anturin normaali käyttöjännite on väliltä 12–24 DCV. Käyttämäni DFM 8D anturi (kuva 10) on siitä erikoinen, että siinä on 2 anturia samassa paketissa, yksi anturi polttoaineen tulolle ja yksi paluulle. Tulon ja paluun erotusta se ei valitettavasti osaa suoraan laskea, vaan anturi lähettää kahta eri signaalia joiden erotus on laskettava systeemin ulkoisesti.



Kuva 10. DFM 8D –virtausanturi.

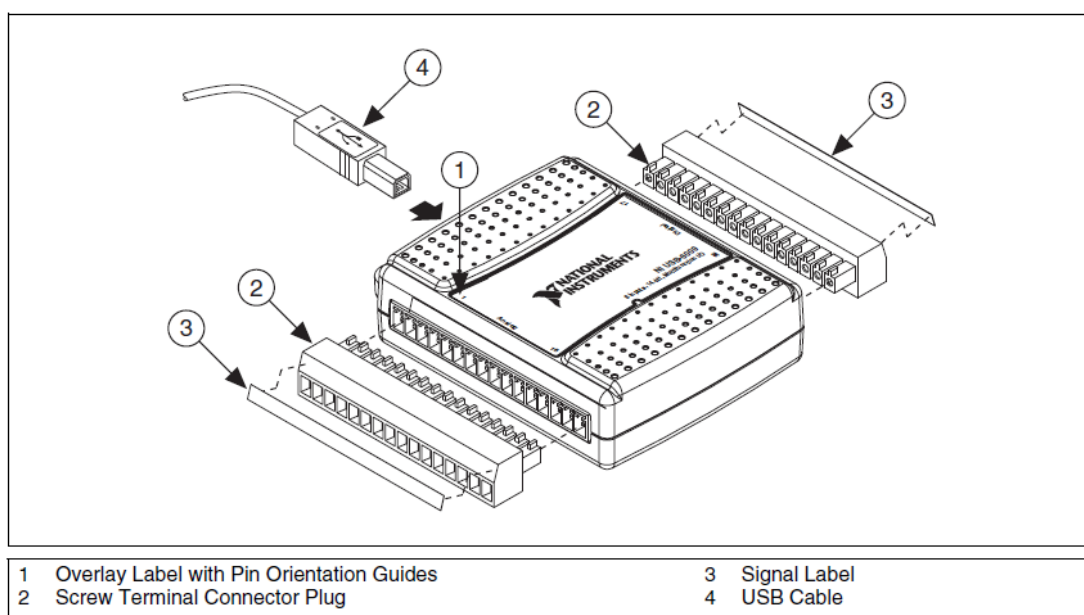
Taulukosta 3 näkyy virtausanturin hydrauliset ominaisuudet, joista havaitaan muun muassa edellisellä sivulla mainittu anturin tilavuus. Lisäksi nähdään, että anturin virhe on hyvin pieni, alle 1 %.





## 6.2 USB-6008 dataloggeri

Dataloggeriksi valikoitui National Instrumentsin USB-6008-perusloggeri (kuva 12) sen monipuolisten ominaisuuksien ja edullisen hinnan vuoksi. Loggeri sisältää 12 analogista sisääntuloa, joista 4 on differentiaalituloa. Lisäksi analogisia ulostuloja on 2 kappaletta ja digitaalisia sisään/ulostuloja 12 kappaletta. Loggerin sisään ja ulostulot näkyvät tarkemmin kuvasta 13. Ainoa loggerin miinus projektiin nähden on sen 10 V:n jännitteen kesto bussin sähköjärjestelmän ollessa 24-volttinen.



Kuva 12. USB-6008 dataloggeri [6, s. 5].

Dataloggerin spesifikaatiot on esitetty taulukossa 5.

Taulukko 5. Dataloggerin ominaisuudet [6, s. 20–23].

General	
Product Name	USB-6008
Product Family	Multifunction Data Acquisition
Operating System	Linux, Mac OS, Pocket PC, Windows
Measurement Type	Voltage
Isolation Type	None
Usb Power	Bus-Powered

<b>Analog Input</b>	
Channels	<b>4, 8</b>
Single-Ended Channels	<b>8</b>
Differential Channels	<b>4</b>
Resolution (bits)	<b>12</b>
Sample Rate (kS/s)	<b>10</b>
Max Voltage (V)	<b>10</b>
Max Voltage Range (V)	<b>-10...+10</b>
Min Voltage Range (V)	<b>-1...+1</b>
Simultaneous Sampling	<b>No</b>
On-Board Memory (B)	<b>512</b>

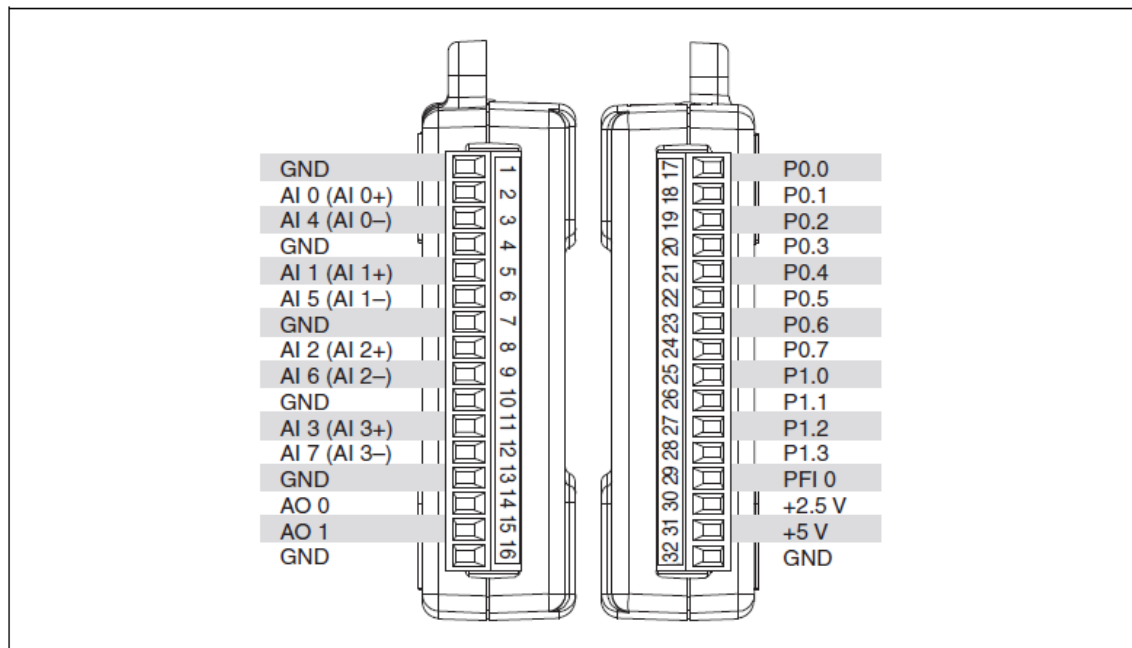
<b>Analog Output</b>	
Channels	<b>2</b>
Resolution (bits)	<b>12</b>
Max Voltage (V)	<b>5</b>
Max Voltage Range (V)	<b>0...5</b>
Min Voltage Range (V)	<b>0...5</b>
Update Rate (S/s)	<b>150</b>
Current Drive Single (mA)	<b>5</b>
Current Drive All (mA)	<b>10</b>

<b>Digital I/O</b>	
Bidirectional Channels	<b>12</b>
Input/Output Only Channels	<b>0</b>
Timing	<b>Software</b>
Logic Levels	<b>TTL</b>
Input Current Flow	<b>Sinking, Sourcing</b>
Output Current Flow	<b>Sinking, Sourcing</b>
Programmable Input Filters	<b>No</b>
Current Drive Single (mA)	<b>8.5</b>
Current Drive All (mA)	<b>102</b>
Maximum Input Range (mA)	<b>0...5</b>
Maximum Output Range (mA)	<b>0...5</b>

<b>Timing/Triggering/Synchronization</b>	
Triggering	<b>Digital</b>
Synchronization Bus (RTSI)	<b>No</b>

Dataloggerissa itsessään ei ole kovalevyä, johon kerätyn datan voisi tallentaa, joten UBS-6008 vaatii ulkoisen laitteen, esim. tässä tapauksessa kannettavan tietokoneen, johon data tallennetaan. Lisäksi USB-6008 saa käyttöjännitteensä USB-väylän kautta.





Kuva 13. Dataloggerin pin-outit [6, s. 9].

### 6.3 NI LabView 2013

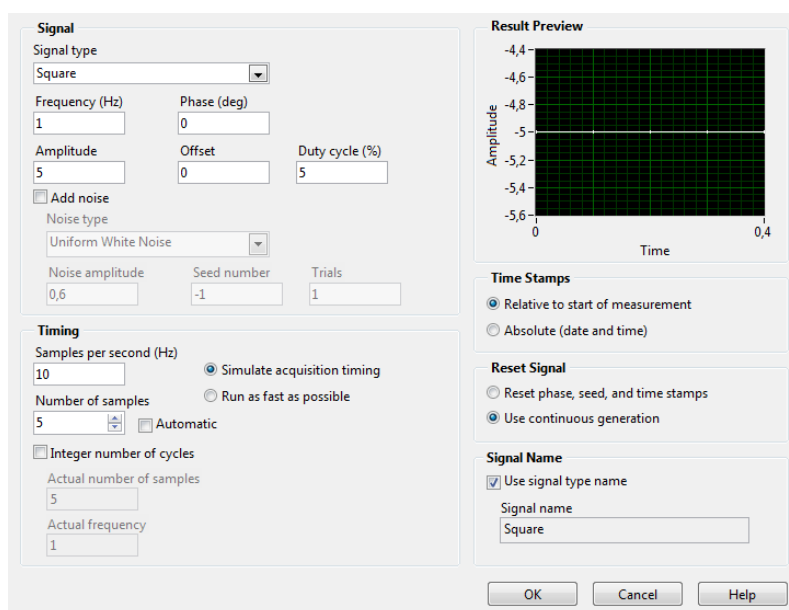
LabView on lyhenne sanoista ”Laboratory Virtual Instrument Engineering Workbench”. LabView 2013 on National Instrumentsin kehittämä alusta ja kehitysympäristö graafiselle ohjelmoinnille. LabView’n ohjelmointikielestä käytetään nimeä G. Alun perin LabView julkaistiin ensimmäisenä Applen Macintoshille vuonna 1986. Uusin versio eli LabView 2013 julkaistiin vuonna 2013 elokuussa. LabView’sta on saatavilla omat versiot Windowsiin, Linuxiin, Maciin, Solarikseen ja joillekin PDA-laitteille. LabView’ssä rakennetut ohjelmat ovat suoraan siirrettävissä eri käyttäjärjestelmien välillä, pois lukien tietyt käyttäjärjestelmäkohtaiset rajoitukset kuten esim. ActiveX-komponentit. LabView on erittäin helppokäyttöinen, ja monipuolisten ominaisuuksiensa ansiosta se soveltuu moniin sovellutuksiin. Yleisimmin sitä käytetään tiedonkeruussa, laitteiston hallinnassa ja teollisuusautomaatiossa. LabView’ssä käytetty G-kieli soveltuu myös yleisohjelmointikieleksi, mutta jotkut operaatiot ovat kuitenkin hyvin työläitä tehdä, esim. mutkittaiden merkkijonojen käsittelyoperaatiot.

## 7 Kulutusmittauksen simulointi

Messuilta tilattuja osia odotellessa simuloin LabView'llä kriittisiä osa-alueita kulutusjärjestelmästä. Simulointi olisi myös etukäteen hyvää harjoitusta ja siitä saisi jonkinlaisen pohjan varsinaiselle datan keruulle ja muunnosohjelmalle.

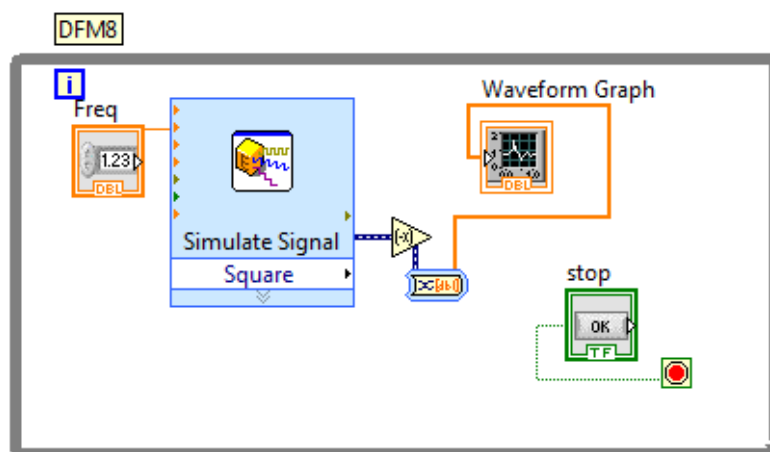
### Vaihe 1:

Ensimmäiseksi oli tärkeintä saada simuloitua virtausanturin antamaa signaalia. Tämä vaihe oli helppo, sillä Labview'stä löytyy suoraan "Simulate Signal" -lohko, jolla saa generoitua useaa eri signaalityyppiä. DFM 8D antaa ulostulona käyttöjännitettään, jonka se maadoittaa, kun tietty tilavuus polttoainetta on virrannut anturin läpi. Toisinsanoen DFM 8D:n ulostulo signaali on kanttiaaltoa. Kuvassa 14 näkyy "Simulate Signalin" konfiguraatioikkuna. Signal typen kohdalla voi valita generoidun signaalin tyyppin ja tässä tapauksessa valittuna on "Square" eli kanttiaalto. Ikkunasta voi säätää myös muita signaaliin liittyviä arvoja, kuten taajuutta, amplitudia, vaihetta jne. Taajuudella ei tässä tapauksessa ole merkitystä, sillä sille luodaan kontrolli, jota voidaan ohjata ohjelmasta käsin. Amplitudiksi valitsin 5 voltia ja "Duty cycle" eli signaalin kahden eri tason väliseksi suhteeksi on valittu 5 %, sillä se vastaa melko hyvin anturin signaali ulostuloa. Alempana kuvassa on määritelty "Timing" asetukset siten, että anturi käyttäytyy mahdollisimman luonnollisesti.



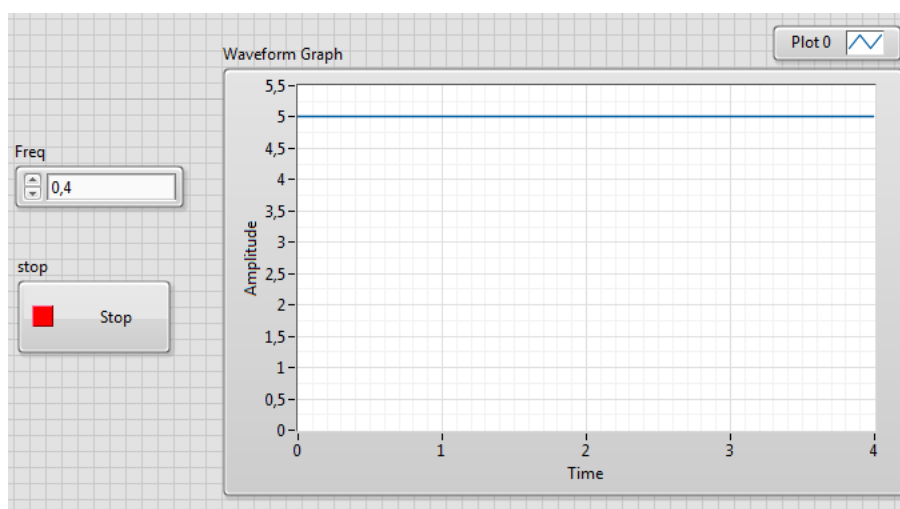
Kuva 14. Simulate signal -lohkon asetukset.

Block Diagrammissa ohjelma näyttää tässä vaiheessa kuvan 15 mukaiselta. "Simulate Signalille" on asetettu yksi kontrolli, ja se on taajuuden säätöä varten. "Simulate Signal"-lohkon ulostulo on johdotettu suoraan "Waveform Graphiin" eli oskilloskoopin ruudulle. Koko ohjelma on laitettu "While-loopin" sisään, jotta se pyörii niin kauan, kunnes stop-nappulaa on painettu.



Kuva 15. Lohkokaavionäkymä ensimmäisen vaiheen jälkeen.

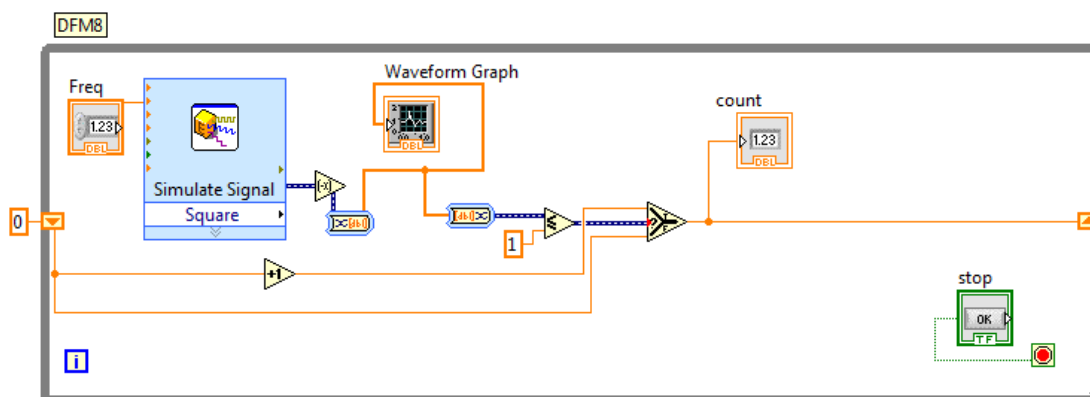
Etupaneelin näkymä on alhaalla olevan kuvan 16 mukainen. Vasemmassa laidassa näkyy taajuuden säätö, ja alempana on ohjelman lopettava stop-nappi.



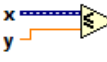
Kuva 16. Etupaneelinäkymä ensimmäisessä vaiheessa.

## Vaihe 2:

Kun anturin signaali on saatu simuloiduksi suhteellisen hyvin, on seuraava askel saada ohjelmamme ymmärtämään sitä halutulla tavalla. Tavoitteena on siis saada ohjelma tunnistamaan, milloin signaali maadoittuu, ja reagoimaan silloin halutulla tavalla.



Kuva 17. Lohkokaavionäkymä toisessa vaiheessa.

Yllä olevassa kuvassa 17 on signaalin ymmärtämiseen vaadittu rakenne. Uutuutena simulaatiomallissa on  eli "Less Or Equal" -lohko, joka yksinkertaisesti vertaa x-arvoa y-arvoon. X:n ollessa pienempi tai yhtäsuuri kuin y, lähettää lohko "true" arvon eli 1:n. Muulloin arvo on "false" eli 0. "Less Or Equal" lohkon x:ään on johdotettu virtausanturin signaali, ja y:n arvoksi on laitettu vakio, tässä tapauksessa luku 1. Toisin sanoen kun virtausanturin 5 voltia maadoittuu alle 1 voltin, tunnistaa ohjelma sen true-arvoksi. Muulloin false-tila on aktiivinen ja "Less Or Equal" -lohko lähettää jokaisella silmukalla 0-arvoa.

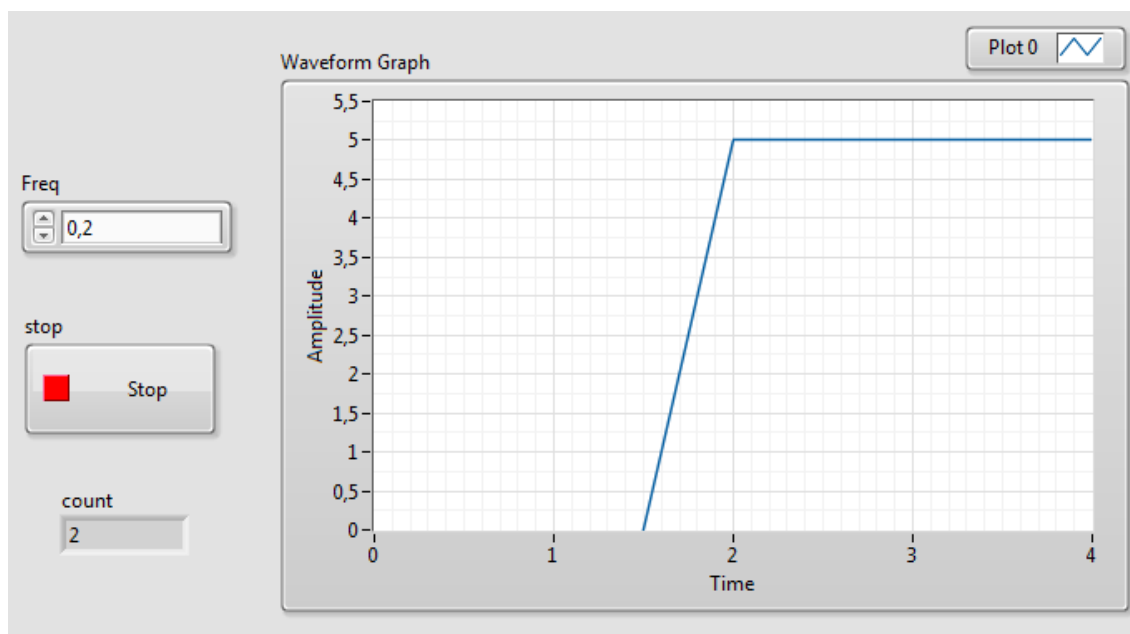
Ohjelmassa uutena lohkona on myös "Count", joka on yksinkertaisesti numeronäyttö, jota ohjelmassa käytetään näyttämään tosi-arvojen lukumäärän. "Less Or Equal" -lohkoa ei kuitenkaan voi suoraan johdottaa näytölle. Näin tehdessä näyttäisi näyttö nimenomaan vain systeemin tilan (tosi/epätosi, 1/0) silmukoittain. Numeronäytössä olisi siis joko arvo 0 tai 1 eikä lukumäärä kasvaisi ollenkaan. Tavoitteena on kuitenkin saada ohjelma laskemaan tosi-arvojen lukumäärä. Tätä varten on käytettävä "Shift Registeriä" eli vietävä yhden silmukan aikana saatu arvo silmukan ulkopuolelle ja palautettava se seuraavaan silmukkaan. Yksinkertaisemmin sanottuna se tarkoittaa siis sitä, että nykyisen silmukan arvo tallennetaan ja palautetaan seuraavaan silmukkaan. Tässä

vaiheessa jos signaaligeneraattorin signaali johdotettaisiin "Count" -lohkoon, näyttäisi se yhden silmukan myöhässä olevaa dataa.



Lopullinen ratkaisu löytyy "Select" -lohkosta . Lohko toimii siten, että keskelle johdotetaan signaali jota halutaan seurata. Signaalin ollessa epätosi eli 0, palauttaa "Select" -lohko alempaan eli f:ään johdotetun arvon. Signaalin ollessa taas tosi eli 1 palauttaa se ylemmän eli t:hen johdotetun arvon. Kun systeemin siis johdottaa edellisellä sivulla olevan kuvan 17 mukaisesti, niin anturimme antaman signaalin ollessa tosi eli maadoittunut, se lisää aina laskuriimme 1. Lisäys tapahtuu "Increment"

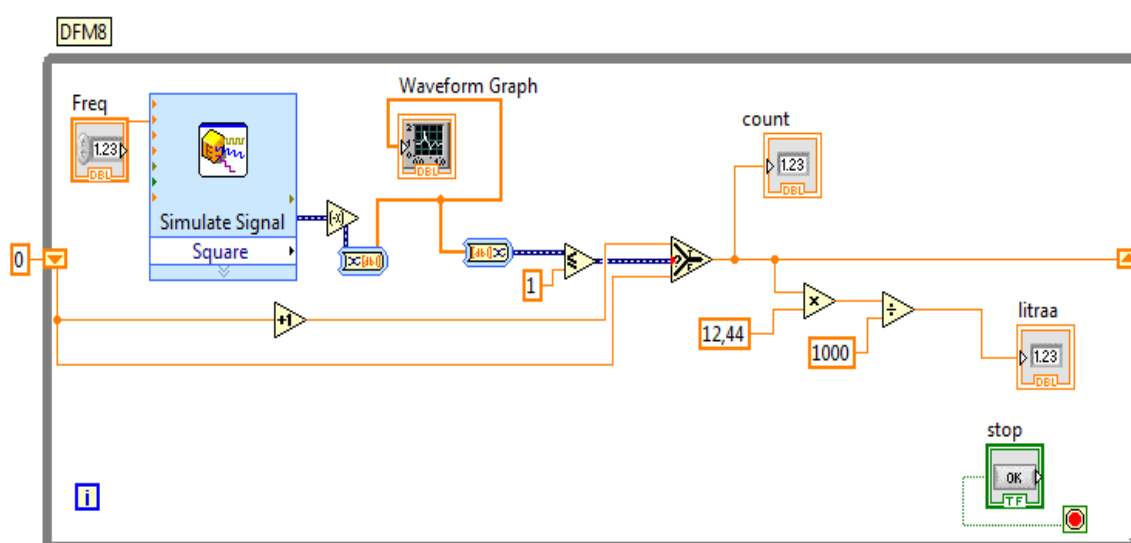
-lohkon  $x \rightarrow x+1$  avulla, joka yksinkertaisesti lisää arvon 1 siihen tulleetseen arvoon. Anturin antaessa 0 tietoa, palauttaa "Shift Register" pelkästään edellisen silmukan arvon koskemattomana. Näin ollen ohjelma ymmärtää, milloin signaali maadoittuu, ja se osaa tuolloin lisätä silmukkaamme arvon 1. Alla on kuva 18, josta näkyy etupaneelin näkymä tässä vaiheessa ja todiste siitä, että signaalin alentua alle 1 voltin, tunnistaa ohjelma sen ja lisää laskuriin arvon 1.



Kuva 18. Etupaneelinäkymä toisessa vaiheessa.

### Vaihe 3:

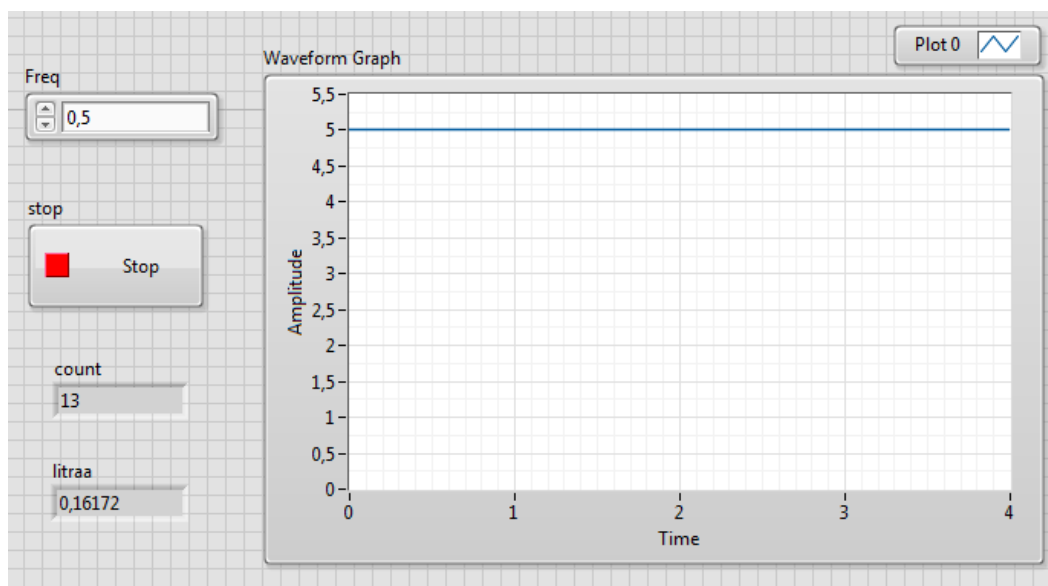
"Count" -lohkon laskemat signaalit eivät ole kovinkaan havainnollisia polttoaineen kulutusta laskettaessa, niinpä ne pitää muuttaa mieluummin esimerkiksi litroiksi. Yleisestihän polttoainetaloudellisuus lasketaan litraa / 100 km, ja niin tässäkin simulaatiossa on tarkoitus, mutta päästäkseen siihen pisteeseen asti, pitää signaalitieto muuttaa ensin litroiksi. Yksi signaalihan anturissa tarkoitti sitä, että sen läpi on virrannut 12,44 ml polttoainetta. Muunnos tapahtuu siis perusmatematiikan avulla. Yksi signaalin arvo kerrotaan ensin 12,44:llä eli muutetaan yksi "Count"-tieto litroiksi. Tämän jälkeen se jaetaan 1000:lla, jotta päästään oikeaan mittayksikköön eli millilitroihin.



Kuva 19. Lohkokaavionäkymä kolmannessa vaiheessa.

Yllä olevassa kuvassa 19 on tämän vaiheen valmis ohjelma. Aikaisemmin mainitut matemaattiset operaatiot ovat toteutettu "Multiply"- ja "Divide"-lohkoilla.

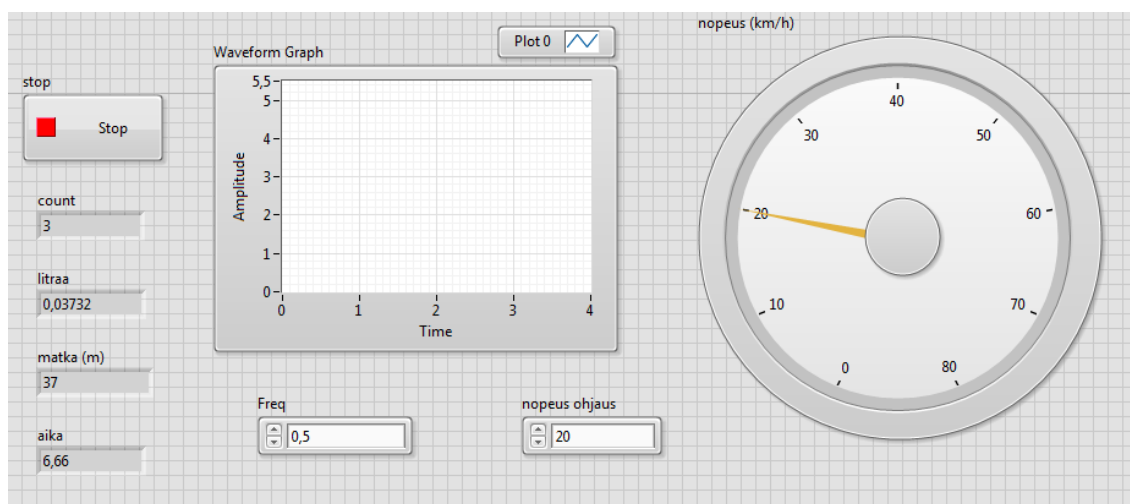
Etupaneeli näyttää tässä vaiheessa seuraavan sivun kuvan 20 mukaiselta. Uutena lisänä etupaneelissa on "litraa"-näyttö, joka kertoo käyttäjälleen virtausanturin läpi virranneen polttoaineen määrän.



Kuva 20. Etupaneelinäkymä vaiheessa kolme.

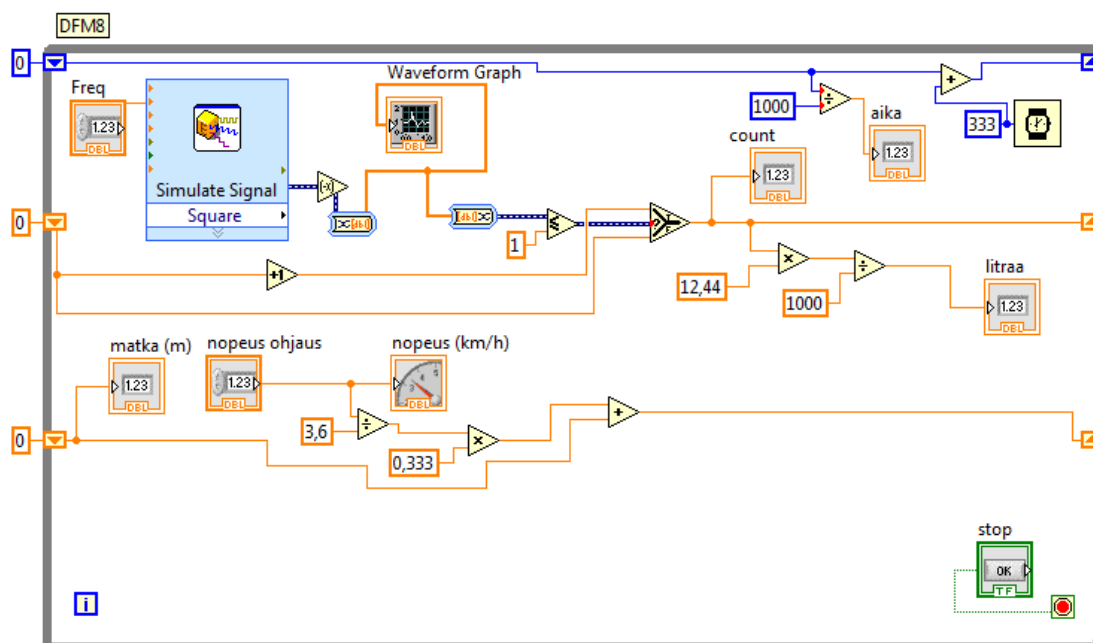
#### Vaihe 4:

Jotta kulutus on mahdollista myöhemmässä vaiheessa saada muotoon litraa / 100 km, on simulaatiossa myös simuloitava matkaa. Alla olevassa kuvassa 21 näkyy ohjelman etupaneelin muutokset. Uutta on nopeuden ohjaus, nopeusmittari, kuljetun matkan ilmaisin ja simulointiajan kertova näyttö.




Kuva 21. Etupaneelinäkymä neljännessä vaiheessa.

Itse ohjelma on kasvanut ja monimutkaistunut aika paljon kuten kuvasta 22 näkee. Nopeuden ohjaus ja mittarin lisäys on aika suoraviivainen toiminto. Ongelmana tässä vaiheessa on kuljettu matka.



Kuva 22. Lohkokaavionäkymä neljännessä vaiheessa.

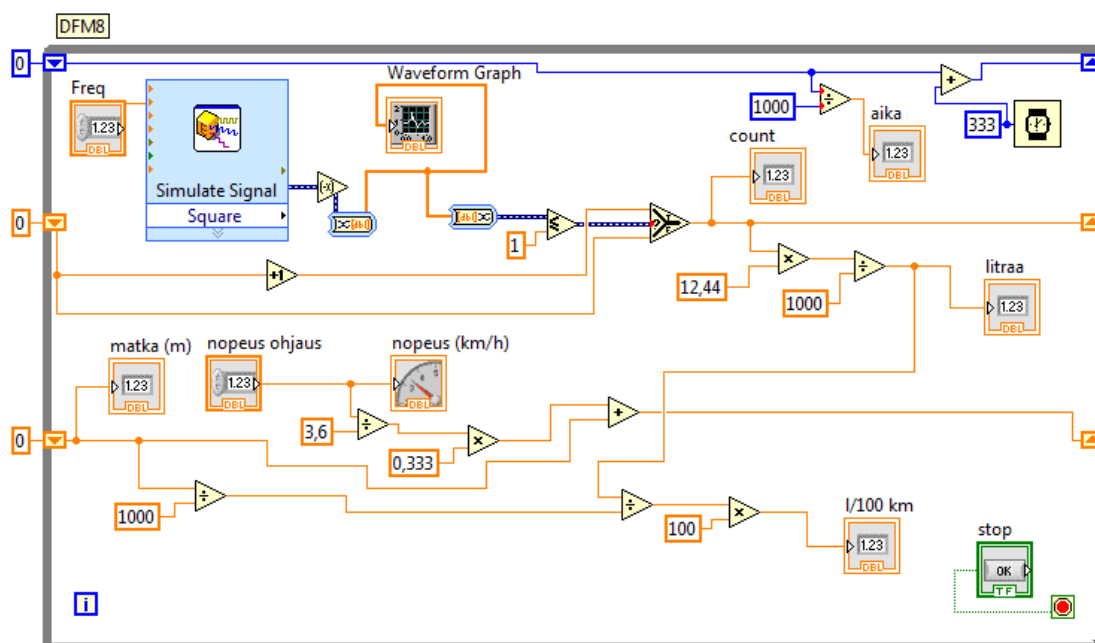
Nopeus säädetään numeerisella ohjaimella ja sen antama tieto johdetaan suoraan mittarille. Nopeus annetaan muodossa km/h, ja jotta myöhemmässä vaiheessa tietoa olisi helpompi käsitellä, on se muutettu muotoon m/s. Oikeassa ylänurkassa on lohko **milliseconds to wait** , joka määrää ohjelmamme päivitystiheyden eli kuinka kauan yksi silmukka kestää. Tähän liittyen myös nopeustieto (m/s) kerrotaan 0,333:lla eli 333 millisekunnilla. 0,333 on itse valittu kiinteä aika, ja se voisi periaatteessa olla mikä tahansa luku. Liian nopea aika kuitenkin saattaa aiheuttaa simulaatiossamme virheitä. Tämä johtuu siitä, että jokainen ohjelmassa näkyvä lohko on tavallaan oma aliohjelmansa. Tämä tarkoittaa taas sitä, että jokainen lohko vaatii hieman aikaa suorittaakseen oman osuutensa. Voidaan puhua siis viiveestä. Yhden lohkon viive on riippuen sen funktiosta n. 5–50 ms. Tässä mallissa viivettä aiheuttavia lohkoja on hyvin vähän joten ohjelman päivitystiheyttä voisi kasvattaa, mutta 333 ms päivitystiheys on aivan riittävä ja säästää myös ohjelmaa pyörittävän prosessorin resursseja.



Nopeuden ohjaimella määräämämme km/h-tieto on ensin muutettu muotoon m/s ja sitten kerrottu ajalla, jolloin tuloksena on luonnollisesti kuljettu matka metreissä. Tämän tiedon voisi johdottaa suoraan matkalle varatulle näytölle, ja se toimisi, mutta ongelmana olisi nopeuden muutokset. Näin johdotettuna jos alkunopeus on esim. 20 km/h ja kesken simulaation se pudotettaisiin nopeuteen 10 km/h, niin myös kuljettu matka putoaisi puoleen. Ohjelma ei siis tavallaan muistaisi jo kuljettua matkaa vaan laskisi matkan pelkästään senhetkisten nopeustietojen perusteella. Siksi on jälleen käytettävä ”Shift Registeriä” ja lisättävä edellisen silmukan matka nykyisen silmukan matkaan. Tällöin jo kuljettu matka säilyy ja pelkästään matkan kasvunopeus suurenee tai pienenee.

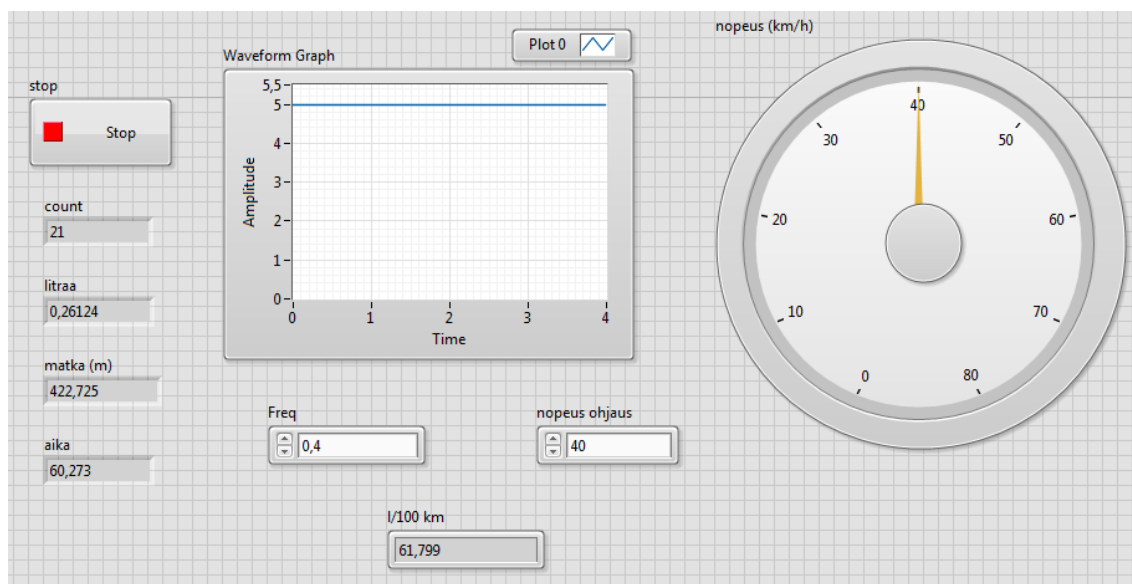
### Vaihe 5:

Viimeisenä tehtävänä on enää saada kulutus muotoon litraa / 100 km. Anturin tunnistama kulutus litroina jaetaan kuljetulla matkalla kilometreiksi muutettuna ja lopuksi kerrotaan vielä sadalla. Ohjelmassa tämä näyttää kuvan 23 mukaiselta.



Kuva 23. Lohkokaavionäkymä viidennessä vaiheessa.

Yllä on siis valmis simulaatiomalli, joka vastaa aika hyvin todellisuutta. Kuvassa 24 taas näkyy valmiin simulaation etupaneeli. Simulaatiota on simuloitu 60 sekuntia vaihtelevilla nopeuksilla ja taajuuksilla. Kulutuslukema on ehkä hieman korkea (todellisuus on n. 40 litraa / 100 km), mutta tässä simulaatiossa anturi toimii hieman nopeammin kuin oikeasti.



Kuva 24. Etupaneelinäkymä viidennessä vaiheessa.

Lopulliseen ”oikeaan” ohjelmaan voi lisätä ominaisuuksia halun ja tarpeen mukaan. Yksi ominaisuus, jota ei simuloitu ja joka ainakin tulee lopulliseen ohjelmaan on mittausdatan tallennus kiintolevylle. Lisäksi oikeassa datankeruuohjelmassa pitää kiinnittää huomiota ohjelman rakenteeseen. Simulaatiomallin rakenne on hyvin monimutkainen, minkä vuoksi se jaksaa päivittyä maksimissaan 100 ms:n välein. Varsinaisen mittausohjelman pitää pystyä päivittymään jopa 20 ms:n välein eli 5 kertaa nopeammin. Pääsyy simulaation hitauteen on virtausanturin signaalin generoiminen. Varsinaisessa ohjelmassa täytyy kuitenkin olla tarkka rakenteen suhteen.

## 8 Järjestelmän asennus

Tässä luvussa käydään läpi fyysisen toteutuksen vaiheet ja lopuksi tarkastellaan saatuja mittaustuloksia.

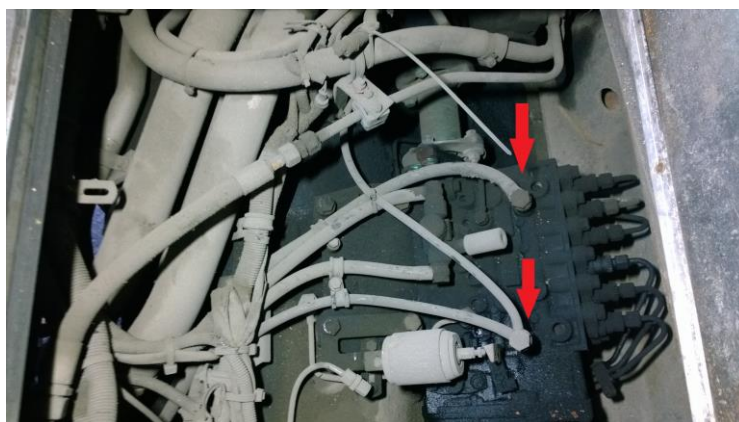
Järjestelmä, kuten jo aikaisemmin on tullut ilmi päätettiin rakentaa PC:n ympärille. Polttoaineen virtausta mitataan virtausanturilla, jonka signaalit PC:hen yhdistettävä dataloggeri kaappaa talteen. PC:hen on tehty LabView'llä käyttöliittymä, joka muuntaa dataloggeriin tulevat signaalit kulutustiedoksi sekä tallentaa mittausdatan PC:n kovalevylle.

Yhtenä ongelmana asennustyössä oli linja-auton pituus. Auto on n. 14 metriä pitkä, ja moottori sekä kuljettajantila ovat käytännössä auton ääripäissä. Tämän vuoksi johdotukset ovat hyvin pitkiä ja tehtävä huolellisesti välttäm. johtojen hankautumista mihinkään.

### 8.1 Virtausanturin asennus

Virtausanturiksi valikoitui Aquametron valmistama DFM 8D -anturi, joka on suunniteltu nimenomaan raskasajoneuvojen kulutusmittauksiin. Se myös kestää huonoja olosuhteita ja esim. moottorin aiheuttamaa värinää. Siksi anturin asennuspaikalla ei ole kovinkaan paljon väliä.

Alla olevassa kuvassa 25 on auton yläpuolisen moottoriluukun näkymä.



Kuva 25. Rivipumppu ja sen polttoaineletkut.

Punaisilla nuolilla merkityt kohteet ovat polttoainelinjan tulo- ja paluukiinnitykset rivipumpussa. Ylempi nuoli on tulo- ja alempi paluulinja. Virtausanturi on asennettava kummankin linjan väliin.



Kuva 26. Virtausanturi asennettuna moottoritilaan.

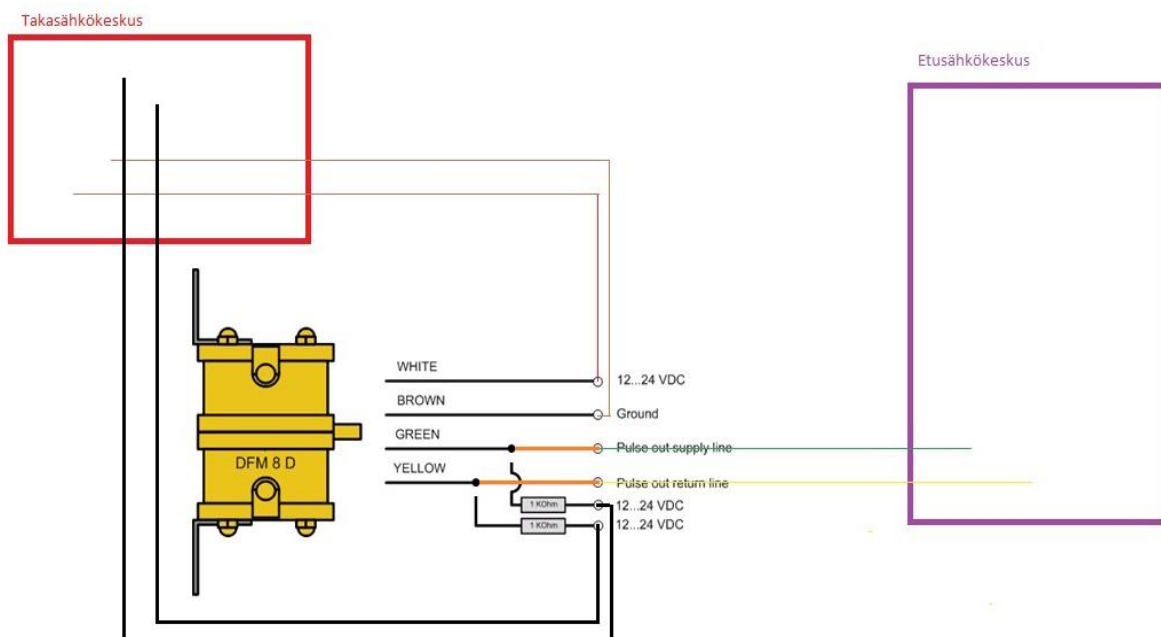
Kuvassa 26 näkyy virtausanturin lopullinen kiinnitys. Anturille tehtiin kiinnikkeet aivan rivipumpun kylkeen, jolloin polttainelinjoja joutui muokkaamaan mahdollisimman vähän. Tulo- ja paluuletkuja lyhennettiin noin 20 cm ja virtausanturilta rivipumpulle tehtiin uudet lyhyet jatkoletkut banjoliittimineen (kuva 27). Asennuksen jälkeen kilometrejä on kertynyt jo lähes 10 000 km, eikä vuotoja ole ollut havaittavissa.



Kuva 27. Anturilta pumpulle tulevat jatkoletkut.

## 8.2 Virtausanturin johdotukset

Alla (kuva 28) näkyy virtausanturin johdotus yksinkertaistettuna. Anturin käyttöjännite (12–24 VDC) otetaan linja-auton takasähkökeskuksesta (kuvan punainen laatikko), jonka kautta mm. käynnistystieto tulee. Anturista lähtevä 4 haarainen johto on vedetty alustaa pitkin takasähkökeskuksen koteloon.



Kuva 28. Virtausanturin johdotuksen havainnekuva.

Plus-jännite eli anturilta lähtevä valkoinen johto on kiinnitetty oranssin nuolen osoittamaan pisteeseen (kuva 29). Kyseisen pisteen jännite on sama kuin akkujännite ja se tulee jännitteelliseksi heti kytkettäessä päävirtakytkin päälle. Valkoista johtoa ei ole kuitenkaan suoraan kytketty plus-pisteeseen, vaan väliin on rakennettu regulaattoripiiri. Piirin tehtävänä on laskea akkujännite (joka auton käydessä saattaa olla jopa yli 28V) 15 volttiin. Reguloinnin syynä oli ensinnäkin se, että anturille on parempi syöttää tasaista käyttöjännitettä. Toisena yhtä tärkeänä syynä oli se, että järjestelmässä käytetyn dataloggerin maksimi jännitekesto on 10V. Jännitettä on siis alennettu valmiiksi jo melkein puoleen, mikä helpottaa jännitteen alentamista vielä ennen dataloggerille tuloa.



Anturin ruskea eli maadoitusjohto on kiinnitetty kuvan 29 punaisen nuolen osoittamaan pisteeseen eli takasähkökeskuksen maadoituslevyyn. Samasta pisteestä on vedetty myös regulaattoripiirille maadoitus.



Kuva 29. Linja-auton takasähkökeskus.

Vihreä (tulo) sekä keltainen (paluu) johto ovat anturin signaalijohdot. Ne ovat liitetty rinnan käyttöjännitteen (15 V) kanssa siten, että välissä kummallakin johdolla on yksi yhden kilo-ohmin ylösvetovastus, kuten kuvassa 28 on piirretty. Signaali-johdot on edelleen johdotettu linja-auton etuosaan auton alustaa pitkin, kuljettajan selän takana olevaan etusähkökeskukseen (kuva 30). Etusähkökeskuksessa sijaitsevat melkein kaikki ajoneuvon tärkeimmät ohjainlaitteet sekä korin ja alustan väliset liitokset. Tämä tarkoittaa sitä, että datan kaappaus kulutusmittaus-järjestelmään on tässä vaiheessa helppoa.

Sähkökeskuksessa virtausanturin signaali jälleen reguloidaan, tällä kertaa 8 volttiin. Tämä jännite soveltuu jo dataloggerille, ja niinpä johdotukset ovat vedetty loppuun siten, että regulaattorilta on tehty 2,5 metrin pituinen johdotus loggerille. Tämä siksi, että johdotus ylettyy hyvin kuljettajantilaan asti tai pyöränkaaren yläpuolella olevaan tavarakaukalo, johon kannettava tietokone on helppo tukevasti asettaa. Dataloggerin ja tietokoneen välinen USB-johto ei ole kuin 30 cm:n pituinen, joten laitteiden on oltava vierekkäin.



Kuva 30. Etusähkökeskus.

### 8.3 Muiden signaalien johdotukset

Jotta ohjelma osaisi laskea kulutuksen muodossa litraa / 100 km, pitää kuljettu matkakin ottaa huomioon. Muita haluttuja ja kulutukseen vaikuttavia mittaustietoja ovat ulkolämpötila ja kaasupolkimen asento.

### 8.3.1 Matkatieto

Matkatieto tulee vaihdelaatikolta mm. etusähkökeskukseen, josta ajoneuvon "Buscom", eli lippujen myyntilaitteistokin tiedon ottaa. Joten itse signaalin kaappaus on helppoa. Signaali on tasaista kanttiaaltoa, jonka jännite vaihtelee 0:n ja 6 voltin välillä sekä taajuus kasvaa nopeuden kasvaessa. Aivan ongelmitta tästäkään vaiheesta ei selvitty, vaan matkatiedon signaalin nopeus eli taajuus tuotti vaikeuksia. Matkatieto, kuten jo aikaisemmin kävi ilmi tulee myös Buscom-laitteistolle, josta on helppo tarkistaa ajoneuvolle määritellyt parametrit. Tietty määrä pulsseja vastaa tiettyä matkaa. Pulssien määrä vaihtelee ajoneuvoittain. Projektin linja-auton pulssimäärä oli 1580 metrin matkalla 11 972. Tämä tarkoittaa sitä, että 1 km:n matkalla pulsseja tulee  $11972/1,58 = 7577,22$  kpl. Ajettaessa esim. 60 km/h eli n. 16,66 m/s, kestäisi 1 km ajaa tasan 60 sekuntia, jolloin pulsseja sekunnissa tulisi  $7577,22/60 = 126$ . Tämä tarkoittaa 130 Hz:n päivitystaajuutta, johon ohjelma ei raskautensa puolesta millään kykene. Jotta pulssia saataisiin hieman rauhoitettua, on dataloggerin ja johtosarjan välissä käytetty pulssimuuntajaa. Sen avulla pulssi on hidastettu 1/16:aan jolloin pulsseja/km tulee 473,58 kpl ja ajettaessa 60 km/h sekunissa 7,89. Tarvittava päivitysnopeus on siis saatu reilulla (20 Hz) varmuuskertoimella runsaaseen 1/6:aan.

### 8.3.2 Ulkolämpötila ja kaasupolkimen asento

Kulutusmittausjärjestelmän ensimmäiseen versioon ei tule ulkolämpötilan ja kaasupolkimen asennon mittausta, vaan kyseiset toiminnot toteutetaan myöhemmässä vaiheessa.

## 8.4 Kulutusmittausohjelman teko ja testaus

Kuten jo aikaisemmin työssä on mainittu, niin mittausdataa käsittelevä ja keräävä ohjelma päätettiin tehdä itse, vaikka markkinoilla olisi ollutkin valmiita paketteja. Ratkaisun suurimmat edut olivat ohjelman sekä käyttöliittymän täysi muokattavuus omiin tarpeisiin. Ohjelma tehtiin nk. graafisella ohjelmoinnilla Labview'n avulla. Tarkemmat tiedot ohjelmasta löytyvät liitteestä 1. Opinnäytetyön tarjoajan pyynnöstä liite on jätetty salaiseksi asiakirjaksi, mutta mittauksien tulokset ja analysointi on esitetty viimeisessä luvussa.



Ensimmäiset testit tehtiin ihan vain varikon pihalla, jotta nähtiin, että virtausanturi antaa edes jotain signaalia. Signaalia tutkittiin myös oskilloskoopin kanssa ja ensimmäisellä testauskerralla huomattiin, että signaalina oli pelkkää kohinaa. Aluksi epäiltiin virtausanturin viallisuutta, mutta virhe löytyi loppujen lopuksi anturin sähköisestä kytkennästä. Anturin signaaliulostulojen ylösvetovastukset olivat unohtuneet asennusvaiheessa. Virheellinen kytkentä kuitenkin havaittiin ja korjattiin, minkä jälkeen signaali oli juuri sellaista kuin piti.

Toinen testi tehtiin jo tien päällä normaalissa liikenteessä, tosin ilman matkustajia ja linja-ajoa. Testeissä huomattiin, että järjestelmän matkalukema ei oikein pidä paikkansa. Signaali tuli kyllä dataloggerille asti, mutta yllätyksenä tuli signaalin käytös. Kuten kulutustietokin, niin myös matkatieto tulee dataloggerille kanttiaaltona (vaihdellen 0:n ja 6 voltin välillä). Yllättävää verrattuna kulutustietoon oli kuitenkin se, että yksi edestakainen pulssi ei ollutkaan yksi signaalitieto vaan pelkkä tilanvaihto tarkoitti yhtä pulssia. Tämän vuoksi myös matka laskettiin ensimmäisellä testauskerralla väärin. Kulutusmittaus ohjelmistoon tehtiin muutos ja toisella testauskerralla matka saatiin laskettua oikein gps:ään verrattuna.

Kolmannella kerralla tapahtui jo oikeata linja-ajoa. Linja-auto ajoi puolikkaan kierroksen linjaa 650, joka kulkee väliä Rautatientori - Ylästö. Tarkoituksena oli ajaa kokonainen kierros, mutta autoon tuli tekninen vika, jonka vuoksi sen piti palata varikolle. Vika ei koskenut mittausjärjestelmää. Kilometrejä kertyi kuitenkin runsaat 30 km. Testi tehtiin siten, että ennen kuin auto lähti linjalle, se tankattiin piri pintaan täyteen. Palatessaan linjalta sama toistettiin ja vertailtiin kulutusjärjestelmän ja tankatun polttoaineen välistä eroa. Testi ei tietenkään ole tieteellisen tarkka, mutta antoi osviittaa siitä, toimiiko kulutusmittaus laisinkaan. Tulokset olivat hieman huolestuttavia. Matkan järjestelmä oli laskenut oikein, mutta itse kulutus oli reilusti alle todellisen lukeman. Ohjelman mukaan kulutus oli noin 32 litraa / 100km, vaikka todellisuudessa se oli ollut runsaat 40 litraa / 100km. Virhe oli sen verran suuri, että jotain oli vialla. Asiaa tutkittiin ja päädyttiin siihen lopputulokseen, että ohjelman rakenne oli liian raskas päivittymään tarpeeksi nopeasti saadakseen kaikki signaalit talteen. Tämän vuoksi ohjelma rakennettiin uudestaan alusta asti ja täysin eri tavalla. Liitteestä 1 selviää tarkemmat muutokset ja uudistukset.

Kun kulutusmittausohjelmaa oli päivitetty rankalla kädellä, oli aika uusia kolmas testivaihe. Tällä kertaa kulutusmittausauto laitettiin samalle linjalle kuin aikaisemmin ja yhdelle kokonaiselle kierrokselle. Sama toistettiin kolme kertaa, kolmena eri päivänä ja

tällä kertaa mitatut ja tankkaamalla lasketut kulutuslukemat olivat noin 0,5 %:n päästä toisistaan. Virheen suuruus oli hyväksyttävissä rajoissa, enkä usko että järjestelmästä paljon tarkempaa edes saisi. Toinen virhettä joko kasvattava tai pienentävä tekijä on tankkausvaihe. Pistooli ei välttämättä ole täysin samassa asennossa kummallakin tankkauskerralla ja tankkausmittarissakin on pieni virhe. Tarkemmat mittaustiedot kerrotaan luvussa 9.

## 9 Mittaukset ja loppupäätelmät

### 9.1 Ensimmäinen testimittaus

Ensimmäisellä varsinaisella testimittauksella ajoin itse tyhjällä linja-autolla suurimmaksi osaksi matka-ajoa pääkaupunkiseudun reuna-alueita pitkin. Tarkoituksena oli saada mahdollisimman paljon kilometrejä ajetuksi lyhyessä ajassa, jotta ensinnäkin nähtiin, kuinka hyvin matkan mittaus toimii ja pysyykö ohjelma perässä vielä korkeissakin nopeuksissa, jolloin pulssisignaali on tiheimmillään. Toinen prioriteetti oli tietenkin tarkistaa, kuinka hyvin kulutusmittaus pitää paikkansa.

Tulokset on esitetty taulukossa 6.

Taulukko 6. Ensimmäisen testimittauksen tulokset.

Mitattu kulutus (l)	Todellinen kulutus	Ero	Virhe %
13,13	16,35	3,22	<b>19,69</b>
Mitattu matka (km)	Todellinen matka	Ero	Virhe %
75,0	74,71	0,29	<b>0,39</b>
Litraa/100km	Todellinen l/100km	Ero	Virhe %
17,51	21,88	4,38	<b>20,00</b>

Kuten taulukosta huomataan, niin matkan mittaus toimii hyvin ja hyvällä tarkkuudella. Kulutuksen mittaamisessa asia on kuitenkin toisin. Virhettä on melkein 20%, ja mitattu kulutus jää yli 3 litraa todellisesta. Johtopäätöksenä oli kuten jo aikaisemmin työssä on mainittu, ettei ohjelma pysy virtausanturin antaman signaalin perässä ja dataa häviää. Toisin sanoen kulutusmittausohjelma oli liian raskas. Alhainen todellinen kulutuslukema

21,88 litraa / 100km johtuu enimmäkseen tasaisesta matka-ajosta ja siitä, että kyseessä oli tyhjänä ajoa ilman mitään kuormaa.

## 9.2 Toinen testimittaus

Toinen mittaus tehtiin ohjelman raskaan päivityksen jälkeen ajaen yksi kokonainen kierros linja-ajoa linjaa 650, joka kulkee Rautatientorin ja Ylästön väliä. Taulukkoon 7 on merkitty kyseisen testimittauksen tulokset.

Taulukko 7. Toisen testimittauksen tulokset.

Mitattu kulutus (l)	Todellinen kulutus	Ero	Virhe %
18,64	18,85	0,21	<b>1,11</b>
Mitattu matka (km)	Todellinen matka	Ero	Virhe %
52,6	52,81	0,21	<b>0,40</b>
Litraa/100km	Todellinen l/100km	Ero	Virhe %
35,44	35,69	0,26	<b>0,72</b>

Tuloksista nähdään heti, että kulutetun polttoaineen mittauksen suuri virhe on saatu päivityksen myötä poistettua ja kokonaisvirhe on laskenut 20 %:sta alle 1 %:iin.

## 9.3 Kolmas testimittaus

Kolmannella kerralla toistettiin toisen kerran onnistunut mittaus, eli linja oli sama 650 ja myös ajon kellonaika oli sama. Taulukossa 8 on mittauksen tulokset.

Taulukko 8. Kolmannen testimittauksen tulokset.

Mitattu kulutus (l)	Todellinen kulutus	Ero	Virhe %
17,63	18,07	0,44	<b>2,43</b>
Mitattu matka (km)	Todellinen matka	Ero	Virhe %
53,4	55,00	1,60	<b>2,91</b>
Litraa/100km	Todellinen l/100km	Ero	Virhe %
33,01	32,85	0,16	<b>0,49</b>

Pieniä eroja toiseen mittaukseen löytyy, ne tosin selittyvät eri kuljettajien poikkeavasta ajotavasta. Eniten tuloksissa silmään pistää kasvaneet kulutuksen ja matkan virheet. Ns. todellinen matka ja kulutus tarkistetaan matkamittarista ja polttoainepumpun mittarista, joissa on oma virheensä ja tämän vuoksi heittoa tuloksissa on

prosenttiyksiköiden verran. Kokonaisvirhe kuitenkin on alle puoli prosenttia, joten järjestelmän tarkkuus on kohtalaisen hyvää tasoa.

#### 9.4 Neljäs testimittaus

Myös neljännellä mittauksella ajettiin sama linja ja samaan kellonaikaan kuin kaksi edellistä mittausta. Taulukko 9 kertoo tämän viimeisen testimittauksen tulokset.

Taulukko 9. Neljännen testimittauksen tulokset.

Mitattu kulutus (l)	Todellinen kulutus	Ero	Virhe %
17,88	18,29	0,41	<b>2,24</b>
Mitattu matka (km)	Todellinen matka	Ero	Virhe %
53,2	54,72	1,52	<b>2,78</b>
Litraa/100km	Todellinen l/100km	Ero	Virhe %
33,61	33,42	0,18	<b>0,55</b>

Toinen mittauskerta eroaa eniten kahdesta jälkimmäisestä (3. ja 4. mittaus). Ensimmäistä mittausta ei oteta huomioon, sillä itse mittausohjelmassa oli silloin ongelmia. Kokonaisvirhe (litraa / 100 km) pysyttelee kuitenkin koko ajan alle 1 prosentin, mikä on aika hyvä tulos. Yksittäiset virheet ovat maksimissaan 3 prosentin luokkaa, mikä tietenkin heikentää mittauksien uskottavuutta. Mutta kuten 8. luvussa on sanottu, todellinen kulutus ja matka perustuvat polttoainepumpun ja matkamittarin lukemaan, joten mahdolliset suuremmat yksittäiset virheet voivat johtua siitä.

#### 9.5 Loppupäätelmät

Useampaa testimittausta ei valitettavasti ehditty eikä pystytty toteuttamaan, mikä tietenkin olisi nostanut mittauksien oikeellisuuden varmuutta, mutta muutaman onnistuneen mittauksen perusteella uskallan väittää, että työ onnistui hyvin. Valmis projekti ei vielä ole, vaan aivan työn alussa mainittuja ominaisuuksia tullaan vielä kokeilemaan ja järjestelmää kokonaisuudessaan vielä parantamaan ja hiomaan. Lisäksi järjestelmää testataan vielä ja varmistetaan sen tarkkuudesta. Tällaisenaan järjestelmä on kuitenkin hyvä pohja ja soveltuu jo oikeisiin mittauksiin pienellä varauksella.

Projektin tekeminen on ollut hyvin opettavaista, sillä se on pakottanut opiskelemaan itselleen täysin uusia asioita (ohjelmointia, polttoainejärjestelmän toimintaa jne.) ja varsinkin projektin alussa olemaan yhteyksissä ja hoitamaan eri asioita eri yritysten kanssa.



## Lähteet

- 1 Veolia Transport Finland. Verkkodokumentti. Veolia Transport Finland. <<http://www.veolia-transport.fi/>>. Luettu 1.8.2014.
- 2 Kuukankorpi, Arttu. 2014. Verkkodokumentti. <<http://www.kuukankorpi.com/paikallisliikenne/liikenneitsijat.html>>. Päivitetty 30.7.2014. Luettu 1.8.2014.
- 3 Volvo Huoltokäsikirja Osa 2, Moottori DH10A.
- 4 Robert Bosch GmbH. 2005. Diesel-Engine Management 4th Edition.
- 5 VZD / VZP and DFM Consumption measurement. 2013. Verkkodokumentti Aquametro AG. Contoil..<[http://www.aquametro.com/downloads/docs/4500e\\_contoil-dfm.pdf](http://www.aquametro.com/downloads/docs/4500e_contoil-dfm.pdf)>. Luettu 4.5.2014.
- 6 User guide and specifications NI USB-6008/6009. Verkkodokumentti. National Instruments Corporation. <<http://www.ni.com/pdf/manuals/371303m.pdf>>. Luettu 20.5.2014